

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平6-326615

(43) 公開日 平成6年(1994)11月25日

(51) Int.Cl. ⁵	識別記号	序内整理番号	F I	技術表示箇所
H 0 3 M 7/40		8522-5J		
G 0 6 F 15/66	3 3 0 A	8420-5L		
G 1 1 B 20/10	3 0 1 Z	7736-5D		
H 0 4 N 5/92	A	4227-5C		
	H	4227-5C		

審査請求 有 請求項の数 4 F D (全 27 頁) 最終頁に続く

(21) 出願番号 特願平5-151595

(22) 出願日 平成5年(1993)5月28日

(31) 優先権主張番号 07/890, 732

(32) 優先日 1992年5月28日

(33) 優先権主張国 米国 (U S)

(71) 出願人 593119446

シーキューブ・マイクロシステムズ・インコーポレイテッド

C-Cube Microsystems, Inc.

アメリカ合衆国カリフォルニア州95035・ミルピタス・マッカーシーブルバード 1778

(72) 発明者 フランク・エイチ・リアオ

アメリカ合衆国カリフォルニア州94087・サニーベイル・ケノーランドウェイ 1655

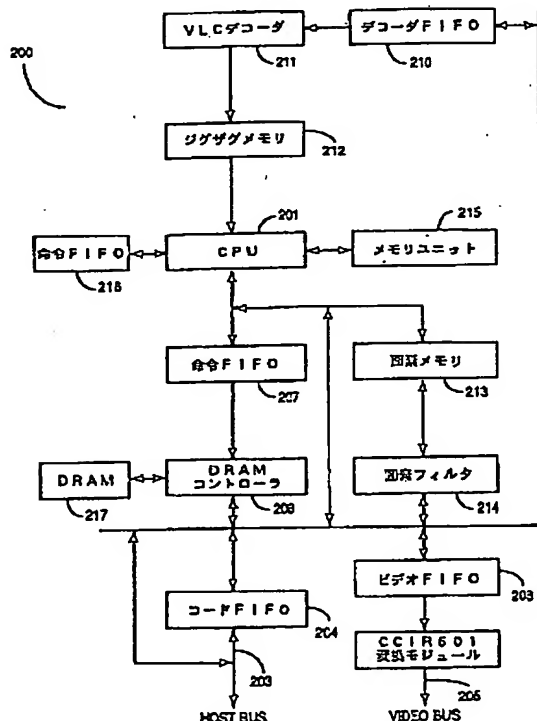
(74) 代理人 弁理士 大島 陽一

(54) 【発明の名称】 可変長コードのコードストリームをデコードする方法及び装置

(57) 【要約】

【目的】 ビデオ信号のリアルタイムの復元をおこない、かつ現在の計算機または集積回路技術を用いた経済的な実施を可能にするべく設計した復元プロセッサを提供することを目的とする。

【構成】 命令を受け取り、かつ命令を特定し、そして特定された命令に基づいて、アドレスの第1フィールドを提供する手段と、コードストリームの始まりのトラックを保持するポインタを提供する手段と、アドレスの第2フィールドとして用いるために、コードストリームの始まりから、予め決められた個数のビットを抽出する手段と、制御情報を記憶するための複数のワードと、各可変長コードのためのデコードされた値とを提供するメモリシステムと、第3フィールドとコードストリームからの予め決められたビットの個数とを用いて、次のアドレスを形成し、メモリシステムへの次のアクセスを起こすための手段とを有する。



【特許請求の範囲】

【請求項1】 中央処理装置からの命令に基づいて、可変長コードのコードストリームをデコードする装置であって、

前記命令を受け取り、かつ前記命令を特定し、そして前記特定された命令に基づいて、アドレスの第1フィールドを提供する手段と、

前記コードストリームの始まりのトラックを保持するポインタを提供する手段と、

前記アドレスの第2フィールドとして用いるために、前記コードストリームの前記始まりから、予め決められた個数のビットを抽出する手段と、

制御情報を記憶するための複数のワードと、前記各可変長コードのためのデコードされた値とを提供するメモリシステムであって、前記メモリシステムの各ワードが、

(1) 前記命令の実行を完了するために、前記メモリシステムへの次のアクセスが要求されているかどうかということと、(2) 前記ポインタを進めるためのビットの個数とを表示し、前記ワードが、前記次のアクセスが必要なことを表示するとき、前記ワードは更に、次のアドレスを生み出すための用いられる第3フィールドを提供し、前記ワードが、前記次のアクセスが必要でないことを表示するとき、前記ワードは更に、前記デコードされた値を提供する、メモリシステムと、前記第3フィールドと前記コードストリームからの前記予め決められたビットの個数とを用いて、前記次のアドレスを形成し、そして前記メモリシステムへの前記次のアクセスを起こすための手段とを有することを特徴とするデコードする装置。

【請求項2】 複数前記可変長コードが、第1の値と、予め決められた値が繰り返される回数を表示する第2の値とをエンコードし、

複数の前記可変長コードの各々のデコーディングが、前記予め決められた値を、前記回数だけ出力することを含み、

後記制御回路を制御するための制御情報を含む、前記複数の可変長コードをデコードする間に、前記予め決められた値と、前記メモリワードとを出力するための制御回路を更に有することを特徴とする請求項1に記載の装置。

【請求項3】 中央処理方法からの命令に基づいて、可変長コードのコードストリームをデコードする方法であって、

前記命令を受け取り、かつ前記命令を特定し、そして前記特定された命令に基づいて、アドレスの第1フィールドを提供する過程と、

前記コードストリームの始まりのトラックを保持するポインタを提供する過程と、

前記アドレスの第2フィールドとして用いるために、前記コードストリームの前記始まりから、予め決められた

個数のビットを抽出する過程と、

制御情報を記憶するための複数のワードと、前記各可変長コードのためのデコードされた値とを有するメモリシステムを提供する過程であって、前記メモリシステムの各ワードが、(1) 前記命令の実行を完了するために、前記メモリシステムへの次のアクセスが要求されているかどうかということと、(2) 前記ポインタを進めるためのビットの個数とを表示し、前記ワードが、前記次のアクセスが必要なことを表示するとき、前記ワードは更に、次のアドレスを生み出すための用いられる第3フィールドを提供し、前記ワードが、前記次のアクセスが必要でないことを表示するとき、前記ワードは更に、前記デコードされた値を提供する、メモリシステムを提供する過程と、

前記第3フィールドと前記コードストリームからの前記予め決められたビットの個数とを用いて、前記次のアドレスを形成し、そして前記アドレスの前記ワードが前記命令の実行が完了したことを表示するまで、前記メモリシステムへの前記次のアクセスを起こすための過程とを有することを特徴とするデコードする方法。

【請求項4】 複数前記可変長コードが、第1の値と、予め決められた値が繰り返される回数を表示する第2の値とをエンコードし、

複数の前記可変長コードの各々のデコーディングが、前記予め決められた値を、前記回数だけ出力することを含み、

後記制御回路を制御するための制御情報を含む、前記複数の可変長コードをデコードする間に、前記予め決められた値と、前記メモリワードとを出力するための制御回路を提供する過程を更に有することを特徴とする請求項3に記載の方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、デジタル信号処理に関する、特にビデオ信号の復元 (decompression) に関する。

【0002】

【従来の技術】 連続的な動きの錯覚を生み出すために、映画では、1秒当たり30コマが提供される。各画像は、数千の画素から作られているので、短いシーンを記憶するために必要なメモリの数でさえも、莫大なものになる。より高い鮮明度の映像が要求されるとき、各画像の画素数もまた増加することが期待される。幸いにも、人間の視覚系の特性を利用することによって、損失のある圧縮技術が、知覚された画質を損なうことなしに、非常に高濃度のデータ圧縮を獲得するべく発達してきた。

(損失のある圧縮技術は、目標の画質を得るためには重要でない情報を捨てることを伴う。) それにもかかわらず、復元プロセッサは、リアルタイムで、記憶されたシーンの各画素を再構成することが要求される。

【0003】モーションピクチャーエキスパートグループ(MPEG)は、圧縮装置と復元装置との間の適合性を得るための規格(以下では、“MPEG規格”と呼ぶ)を提供した。この規格は、記憶媒体のためのビデオ信号のデコードされたデジタル表現とデコードの方法との両方を明記している。その表現は、カラー映画のその他の上映方法やスチール写真の複写と同様に、正規の速度の再生を支持する。その規格は、一般的な525本及び625本のラインのテレビ、パーソナルコンピュータ及びワークステーションのディスプレイのフォーマットを包含する。MPEG規格は、コンパクトディスク、デジタルオーディオテープ、及び磁気ハードディスクのような、1秒当たり1.5メガビットの連続的な転送速度を支持する装置を目的としている。MPEG規格は、24Hzから30Hzの間の周波数での略288×352画素の映像フレームを支持することを目的としている。国際標準化機構(ISO)による刊行物“Coding for Moving Pictures and Associated Audio--for digital storage media at up to about 1.5Mbit/s,”は、草案の形で提案されたMPEG規格を提供している。MPEG規格に関する詳細な情報を提供するために、この刊行物は、ここで言及したことによって本出願の一部とされたい。

【0004】MPEG規格では、映像フレームは、一連の“マクロブロックスライス”(MBS)に分割され、各MBSは、各々が16×16画素の領域を包含する(マクロブロックと呼ばれる)複数の画像領域を含んでいる。これらの画像領域の各々は、その要素が空間的な輝度の値及び色光度の値からなる1個または複数の8×8マトリクスによって表現される。マクロブロックのある表現(4:2:0)では、輝度の値(Yタイプ)は、16×16画素の画像領域内(4個の8×8Yマトリクス内)の各画素のために提供され、各々が同一の16×16画像領域を包含するU及びVタイプの色光度(即ち、青及び赤の色光度)が、8×8“U”マトリクス及び8×8“V”マトリクスに提供される。即ち、各8×8“U”マトリクスまたは8×8“V”マトリクスが、16×16画素の領域を包含する。他の表現(4:2:2)では、輝度の値は、16×16画素の画像領域内の各画素に提供され、U及びVタイプの各々に対する2個の8×8マトリクスが、16×16画素の画像領域の色光度を表現するべく提供される。

【0005】MPEG規格は、図1に示された圧縮及び復元のモデルを採用している。図1に示すように、最初に(ブロック101で表現された)フレーム間の冗長性が、カラー映画のコマから除去される。フレーム間の冗長性を除去するには、各フレームが、デコードのために、“内部の(intra)”、“予測された(predicted)”及び“内挿された(interpolated)”の内の何れかの名前を与えられる。内部のフレームは最も少なく提

供され、予測されたフレームは内部のフレームよりも多く提供され、そして残りの全てのフレームが内挿されたフレームである。内部のフレーム(“Iピクチャー”)内の圧縮されたビデオデータは、内部のフレーム内の画素のみから計算される。予測されたフレーム(“Pピクチャー”)内では、最新のIピクチャーまたはPピクチャーからの画素の値の増加だけがデコードされる。内挿されたフレーム(“Bピクチャー”)内では、最初のフレームとの最後のフレームの両方に関して、画素の値がデコードされる。フレームを増加したときだけデコードし、予測されたフレーム及び内挿されたフレームを用いることによって、多くのフレーム間の冗長性が除去され、メモリを大量に節約することができる。全体のマクロブロックの動作は、画素レベルではなく、動作ベクトルによってデコードされ、それによって更なるデータ圧縮が提供される。

【0006】MPEG規格のもとでの圧縮のステップは、内部フレームの冗長性を除去する。図1のブロック102で表される第1ステップでは、二次元の離散コサイン変換が、空間的な輝度の値及び色光度の値を周波数領域に写すために、8×8数値マトリクスの各々で実行される。

【0007】図1のブロック103で表される次のステップでは、“量子化”と呼ばれる過程が、8×8マトリクスの各要素の輝度または色光度、及び周波数に基づいて各要素に重み付けをする。量子化の重み付けは、人間の目が感知できない高周波成分を0に減少させることを目的としている。8×8行列内に多くの0要素を生み出すことにより、各マトリクスは、情報の損失なしに、“DC”値、及び0でない“AC”値と、それに続く0の値の長さとの交互の一对の順序リストとして表現される。順序リストは、マトリクスがジグザク型に読まれる(即ち、マトリクスAの要素が、A(0,0)、A(0,1)、A(1,0)、A(0,2)、A(1,1)、A(2,0)、という順序で読まれる)べく、マトリクスの要素が存在するように順序づけられている。この表現は、0の要素を個々に表現していないので、空間を節約する。

【0008】最後に、図1のブロック104で表されたエントロピーデコード方式が、可変長コードによって、更に、DCブロック係数と、非0のAC値と0の要素の長さとの交互の一对を圧縮するために用いられる。エントロピーデコードでは、より頻繁に発生するシンボルほど、より短い符号によって表現される。従って、メモリ空間が更に節約される。

【0009】MPEGでの復元が、図1のブロック105~108によって表されている。復元では、エントロピーデコード、量子化及びDCTの過程が、ブロック105~107に各々表されたように、入力と出力が逆にされる。“実際の画素の発生”と呼ばれる最後のステッ

ブ(ブロック108)は、プレイモード(順方向、逆方向、スローモーション等)、及び使用されているディスプレイの寸法及び特質に基づいて、再生のための実際の画素を提供する。

【0010】更に、MPEG規格は、インターレースされていないビデオ信号のためだけに提供されているので、従来のNTSCテレビまたはPALテレビに出力映像を表示するために、復元器(decompressor)は、従来のインターレースされたフィールド内の出力ビデオ信号を提供しなければならない。インターレースされたビデオ信号の復元のためのガイドラインは、MPEG規格の拡張部分として提案されている。この拡張された規格は、国際ラジオ諮問委員会(CCIR)の勧告601(CCIR/601)と両立する。

【0011】上述されたMPEG規格に対して例示された圧縮及び復元に含まれるステップは、計算機にとって非常に集約的なもので、そのような圧縮方式が実際にかつ広く受け入れられるためには、復元プロセッサは、リアルタイムの復元を提供し、かつ現在の計算機または集積回路技術を用いた経済的な実施を可能にするべく設計されなければならない。

【0012】

【発明が解決しようとする課題】本発明の目的は、ビデオ信号のリアルタイムの復元をおこない、かつ現在の計算機または集積回路技術を用いた経済的な実施を可能にするべく設計した復元プロセッサを提供することである。

【0013】

【課題を解決するための手段】上述した目的は、中央処理装置からの命令に基づいて、可変長コードのコードストリームをデコードする装置であって、前記命令を受け取り、かつ前記命令を特定し、そして前記特定された命令に基づいて、アドレスの第1フィールドを提供する手段と、前記コードストリームの始まりのトラックを保持するポインタを提供する手段と、前記アドレスの第2フィールドとして用いるために、前記コードストリームの前記始まりから、予め決められた個数のビットを抽出する手段と、

【0014】制御情報を記憶するための複数のワードと、前記各可変長コードのためのデコードされた値とを提供するメモリシステムであって、前記メモリシステムの各ワードが、(1)前記命令の実行を完了するために、前記メモリシステムへの次のアクセスが要求されているかどうかということと、(2)前記ポインタを進めるためのビットの個数とを表示し、前記ワードが、前記次のアクセスが必要なことを表示するとき、前記ワードは更に、次のアドレスを生み出すための用いられる第3フィールドを提供し、前記ワードが、前記次のアクセスが必要でないことを表示するとき、前記ワードは更に、前記デコードされた値を提供する、メモリシステムと、

前記第3フィールドと前記コードストリームからの前記予め決められたビットの個数とを用いて、前記次のアドレスを形成し、そして前記メモリシステムへの前記次のアクセスを起こすための手段とを有することを特徴とするデコードする装置と、中央処理方法からの命令に基づいて、可変長コードのコードストリームをデコードする方法であって、前記命令を受け取り、かつ前記命令を特定し、そして前記特定された命令に基づいて、アドレスの第1フィールドを提供する過程と、前記コードストリームの始まりのトラックを保持するポインタを提供する過程と、前記アドレスの第2フィールドとして用いるために、前記コードストリームの前記始まりから、予め決められた個数のビットを抽出する過程と、制御情報を記憶するための複数のワードと、前記各可変長コードのためのデコードされた値とを有するメモリシステムを提供する過程であって、前記メモリシステムの各ワードが、

(1)前記命令の実行を完了するために、前記メモリシステムへの次のアクセスが要求されているかどうかということと、(2)前記ポインタを進めるためのビットの個数とを表示し、前記ワードが、前記次のアクセスが必要なことを表示するとき、前記ワードは更に、次のアドレスを生み出すための用いられる第3フィールドを提供し、前記ワードが、前記次のアクセスが必要でないことを表示するとき、前記ワードは更に、前記デコードされた値を提供する、メモリシステムを提供する過程と、前記第3フィールドと前記コードストリームからの前記予め決められたビットの個数とを用いて、前記次のアドレスを形成し、そして前記アドレスの前記ワードが前記命令の実行が完了したことを表示するまで、前記メモリシステムへの前記次のアクセスを起こすための過程とを有することを特徴とするデコードする方法とを提供することによって達成される。

【0015】

【作用】本発明に基づけば、可変長コードとしてエンコードされた、圧縮された離散コサイン変換(DCT)係数のデコーディングを提供する装置及び方法が提供される。

【0016】第1実施例では、前記装置は、グローバルバスを通しての通信を行う複数のコプロセッサユニットを制御するマイクロ中央処理装置を有する。コプロセッサユニットは、(1)可変長コードのストリームを受け取るためのホストバスインタフェースと、(2)受け取られた可変長コードのストリームを記憶し、検索するための外部ランダムアクセスメモリを制御するためのメモリコントローラと、(3)圧縮された可変長コードをDCT係数に変換するための復元器及び検出器と、(4)DCT係数を画素の数値に変換するための、逆離散コサイン変換プロセッサと、(5)画素の値を再サンプリングし、かつひと続きの画面の基準(内部)フレーム内の情報に基づいた、デコードされたひと続きの画面を再構成

するための画素フィルタ及び動作補償装置とを有する。

【0017】本発明の他の技術的視点に基づけば、量子化機能及び逆コサイン変換機能は、中央処理装置内の特別な目的のハードウェアによって実行される。更に、逆コサイン変換機能は、(1)第3データよりも小さい、第1データと第2データの和に等しい結果を計算するために、第1、第2、及び第3データをオペランドとして受け取る第1ステージと、(2)第1ステージから結果を受け取り、第4データと第1ステージからの結果との和及び差の両方を計算する第2ステージとを有する構造によって実行される。ある実施例では、第1、第2及び第3データがレジスタファイルから得られ、中央処理装置が、第1ステージからの結果がレジスタファイルへ戻らないように(バイパスしないように)指図するとき以外は、第1ステージ及び第2ステージの結果がレジスタファイルに戻される。

【0018】本発明の他の技術的視点に基づけば、メモリシステムを制御し、かつ複数のコプロセッサユニットに役立つメモリコントローラは、各コプロセッサユニットに対して、ファーストインファーストアウト(FIFO)メモリを配置し、関連するコプロセッサユニットへのメモリアクセスリクエストを別々の待ち行列に入れる。メモリコントローラ内の優先順位回路は、予め決められた優先順位方式のもとで、待ち行列内の最も高い優先順位を有するメモリリクエストに対してメモリがアクセスすることを許容する。メモリシステムへの複数のアクセスを要求するメモリアクセスリクエストに対しては、その複数のアクセスが完了する前に、メモリコントローラにより高い優先順位のメモリアクセスリクエストが到達した場合、その複数のメモリアクセスリクエストは、より高い優先順位のメモリアクセスリクエストに先んじられる。

【0019】本発明の他の技術的視点に基づけば、復元器及びデコーダによる可変長コードのデコーディングは、可変長コードのデコードされた値をも記憶する、読み出し専用メモリのような制御メモリシステム内のアクセスされたメモリワードの内容によって制御される。最初に、復元器及びデコーダは、コードストリームからの予め決められたビット数及び中央処理装置から受け取られた命令に基づく予め決められたビットパターンによって形成されたアドレスを用いて、制御メモリシステムをアクセスする。次に、制御メモリシステム内のアクセスされたワードは、更なるメモリアクセスが要求されているかどうかを決定するために用いられる。即ち、アクセスされたワードは、可変長コードのデコードされた値または制御情報、またはその両方を含む。もし、制御メモリシステムへの更なるアクセスが必要ならば、コードストリームから得られた予め決められたビット数と、制御メモリシステム内の最後にアクセスされたワードの内容の一部とを用いて、新しいアクセスが完成される。可変

長コード(“ラン長さ”)が0の値の数をデコードするある実施例では、このデコーディング方法による、制御メモリシステムのアクセスされたワードは、これらの0の値の出力を制御する。

【0020】

【実施例】本発明は、以下の詳細な説明と添付の図面とによってよりよく理解される。

【0021】概説

本発明の実施例のブロック図が図2に示されている。図2では、ビデオデコーダ200は、中央処理装置(CPU)201と、ホストバス203によるホスト計算機202(図示されていない)とのインタフェースとを有する。ホスト計算機202は、ビデオデコーダ200によってFIFOメモリ204(“コードFIFO”)内に受け取られる圧縮されたビデオデータのストリームを提供する。ホスト計算機202から受け取られた圧縮されたデータは、ビデオデコーダ200によって復元され、そして復元されたビデオデータは、ビデオデコーダ200の出力データとして、ビデオバス205に提供される。

【0022】ビデオデコーダ200のCPU201は、制御記憶装置(“命令メモリ”)216を有するマイクロプロセッサである。CPU201は、FIFOメモリ207(“命令FIFO”)を通して、メモリモジュール217(“DRAM”)を制御するメモリコントローラ206(“DRAMコントローラ”)に命令を送る。この実施例では、DRAM217は随時読み出し書き込み可能なメモリ要素からなるが、しかし、記憶装置を実施するための他の適切なメモリ技術が用いられてもよい。DRAM217は、ホスト計算機202から受け取られた圧縮されたデータと、ビデオバス205への出力のための復元されたデータとの両方を記憶する。ビデオバス205への出力のための復元されたデータは、出力FIFOメモリ208(“ビデオFIFO”)内で待ち行列に入れられる。

【0023】この実施例では、ビデオプロセッサ200の動作モジュールは、グローバルバス209を通して連絡している。グローバルバス209の制御は、優先順位方式のもとで、DRAMコントローラ206、ホスト計算機202、及びCPU201の何れかに対して行われる。動作の間、ホスト計算機202から受け取られた圧縮されたビデオデータは、CPU201の命令のもとで、DRAMコントローラ206によってDRAM217に記憶される。この圧縮されたデータは、復元のために、CPU201の命令のもとでDRAM217から検索され、FIFOメモリ210(“デコーダFIFO”)を通して、可変長コード(VLC)デコーダ211に転送される。MPEG規格に基づけば、復元されたデータは、最初にメモリ212内に“ジグザグ”順に記憶されることによって、整理し直され、次にジグザグメ

メモリ212から、列の大きい順に検索される。列の大きい順の圧縮されたデータは次にCPU201に提供され、CPU201では、その圧縮されたデータは、“逆量子化”され、二次元の逆離散コサイン変換(IDCT)によって変換される。IDCTは、圧縮されたDCT係数を、周波数領域の表現から空間領域(“画素空間”)の表現へ変化する。逆量子化動作及びIDCT動作を実行するとき、CPU201は、ローカルメモリ逆量子化、コサイン、及び他の定数から検索を行う。二次元IDCTの中間の結果を記憶するための一時記憶装置もまた、記憶装置215によって提供される。記憶装置215は、量子化メモリ215a、一時記憶装置215b、及びコサインメモリ215cからなる。逆量子化動作及びIDCT動作は、以下に、より詳しく説明される。

【0024】画素空間の圧縮されたデータは、FIFOメモリ213(“画素メモリ”)内に記憶される。これらの画素空間のデータ(“画素”)は、画素フィルタ214によってフィルタされ、かつ“動作補償”される。フィルタリング及び動作補償の動作は、以下に更に詳しく説明される。CPU201の指令によって、DRAMコントローラ206は、画素フィルタ214によって動作補償された画素をDRAM217内に記憶する。CPU201の指令によって、次に、画素がDRAM217からDRAMコントローラ206によって検索され、ビデオデコーダ200の出力データとして、グローバルバス209を通してビデオFIFO208に提供される。CCIR601変換モジュールは、ユーザが選択できるオプションとして、圧縮された出力ビデオデータのCCIR601フォーマットに適合したビデオデータへの交換をおこなう。本実施例では、ユーザは、1秒当たり30フレーム速度での352×240の映像、または1秒当たり60フレームのフレーム速度での704×240の映像(即ち、CCIR601フォーマット)を選択できる。CCIR601への変換は、水平内挿技術及びフレーム速度変換技術の両方によって達成される。

【0025】内部グローバルバス

グローバルバス209は、3個のソースである、ホスト計算機202、CPU201、及びDRAMコントローラ206からドライブされている。内部グローバルバスは、8ビットアドレスバスGSEL209a及び16ビットデータバスGDAT209bからなる。グローバルバス209をアクセスする2クロック周期前に、アクセスを要求する装置がリクエストビットをアクティブ状態にする。予め決められた優先順位方式に基づいて、次のクロック周期内で、最も高い優先順位を有する、アクセスをリクエストしている装置が、データの送信または受信を行おうとするモジュールのアドレスをアドレスバスGSEL209a上にドライブする。別々のGSEL

アドレスバスGSEL209a上のアドレスが提供された後の1クロック周期内で、データのソース(即ち、書き込み動作中では、アクセスをリクエスト中の装置、あるいは、読み出し動作中では、アクセスされたモジュール)によって、データがデータバスGDAT209a上にドライブされる。

【0026】この実施例では、DRAM217へのアクセス毎に、2クロック周期が必要なので、データがDRAMコントローラ206に書き込まれるかまたはDRAMコントローラ206からリクエストされる最高速度は、他の1クロック周期である。

【0027】ホストバスインタフェース

本実施例では、コードFIFO204は、2バイト幅であって、かつ32ビットの圧縮されたコードを保持する。ホストバスインタフェース203は、20ビットアドレスバス203a、16ビットデータバス203b、及びデータの転送を実行するか、信号を送るための制御バスを有する。ホストバスインタフェース203は、ホストプロセッサクロック、チップクロック、システムクロック、アドレス有効ストローブ(AS)信号、読み出し/書き込み(R/W)信号、データレディ信号UDSとLDS、リセット信号、及びテスト信号を含む。ホスト計算機202は、DMAモードでのコードFIFO204内への書き込みによって、圧縮されたビデオデータを転送する。圧縮されたビデオデータは、DMAモードで、バス書き込みサイクル当たり16ビットの割合で、16ビットデータバス203b上に転送される。非DMA転送は、制御機能を実行し、そしてコードFIFO204をアクセスするため、及びDRAMコントローラ206を通してDRAM217をアクセスするために、ホスト計算機202によって利用される。

【0028】幾つかの制御レジスタが、制御機能を実行するためにホストバス202によってアクセスされる。これらのレジスタは、(1)ホスト計算機202が、DMA制御レジスタをイネーブルまたはディスエーブルして、FIFO204をコーディングすること、かつFIFO204の状態を照会することを許可するDMA制御レジスタと、(2)ビデオデコーダ200が、ホスト計算機上でベクトルインタラプトを実行することを許可するベクトルインタラプトレジスタと、(3)ホスト計算機202が、圧縮されたビデオデータストリームを、音声部分のような他のMPEGデータストリームと同期させることを許可するシステムタイマレジスタとを含む。

【0029】非DMAモードで、これらの制御レジスタの任意の1個、DRAM217、及びコードFIFO204の何れかをアクセスするために、ホスト計算機202は、R/W信号を適切にセットする“AS”を発生させ、かつ20ビットアドレスバスにアドレスを配置する。20ビットアドレスのビット(20:19)と、R/W信号とは各々アクセスの行き先とアクセス形式を表

示している。書き込みアクセスのために、ホスト計算機202は、転送されるべきデータを16ビットデータバスに配置し、データレディ信号UDS及びLDSを立ち上げる。それに対して、ビデオデコーダ200は、16ビットデータをラッチ、そして応答し、それによってホストバス書き込みサイクルが完了する。読み出しアクセスのために、ビデオデコーダ200は、リクエストされたデータが16ビットデータバス203bでレディ状態になったとき、AS信号に応答する。

【0030】中央処理装置

CPU201は、24ビットデータバスと32個の汎用レジスタ(“レジスタファイル”)とを有するマイクロプロセッサである。CPU201は、コプロセッサ例えばメモリコントローラ206を制御するだけでなく、動作補償(後の節で説明される)のための最初のアドレスを計算したり、量子化及びIDCTを実行する。以下に説明されるように、汎用のハードウェアと特定の目的のハードウェアの両方がCPU201に備えられている。IDCT及び非IDCTの計算の両方によって用いられる汎用のハードウェアは、レジスタファイルと、乗数を
10 含む算術論理演算装置(ALU)とを有する。逆量子化及びIDCTの計算で用いられる特定の目的のハードウェアは、5×8乗算減算装置601、“バタフライ”装置602、コサイン読み出し専用メモリ(ROM)215c及び量子化メモリ215dを有する。

【0031】CPU201は、MPEG規格に明記された逆量子化動作及びIDCT動作のための特別な表示を提供する。特に、3つの乗算命令がこれらの動作のための設計されている。各乗算命令はまた、同時に“バタフ
15 ライ”計算をも実行する。当業者には公知のバタフライ計算は、2つの数の和と差を同時に計算する。バタフライ計算は、しばしばデジタルフィルタ内で用いられる。

【0032】Alexandre Balkanskiらによって、1990年3月14日に出願された米国特許出願第07/494,242号明細書“A system for Compression and Decompression of Video Data Using Discrete Cosine Transform and Coding Techniques”に開示された8×8画素2次元IDCTアルゴリズムに基づいて、CPU201は、IDCT動作を実行するようにプログラムされている。この明細書は、ここで言及したことによって本
20 出願の一部とされたい。IDCT動作は、DCT係数の8×8ブロックまたはマトリクスでの列と列または行と行の何れかの2個の一次元IDCT動作の実行によって行われる。図5は、8×8ブロックの1列または1行で動作するために利用される8ポイントのIDCTアルゴリズムの流れ図である。図5から推論できるように、DCT係数の各列または各行に対して、コサイン因子による13個の乗算及び12個のバタフライ動作が実行される。図5では、表示法C3/16は、コサイン因子COS(3π/16)による乗算を表し、ここでπは円周率
25

3.14159…を表している。

【0033】DCT係数の逆量子化が、MPEG規格に基づいて実行される。逆量子化係数は、量子化メモリ(“QMEM”)215aに記憶されている。逆量子化は、8×8マトリクスの各DCT係数に、対応する逆量子化定数を乗算することによって達成される。

【0034】逆量子化及びIDCT動作のデータフローは、図6にまとめられている。図6に示すように、DCT係数の8×8ブロックの列を構成する8個の9ビットDCT係数は、同時に、ステップ501でジグザグメモリ212から検索される。これらのDCT係数は、ステップ502で逆量子化され、ステップ504でのDCT係数上の最初の一次元IDCTの実行の前に、ステップ503で適切なコサイン因子によって乗算される。結果として得られた8個のDCT係数は、次に一次元IDCTが8×8ブロックの各列で完了するまで、ステップ505で一時記憶装置215bに記憶される。次に、8×8ブロックのDCT係数の行を形成する8個のDCT係数は、同時に、2番目の一次元IDCT動作のために、
30 ステップ506で検索される。ステップ507では、2番目の一次元IDCT動作からの結果として得られた画素の値が、画素メモリ213に提供される。

【0035】一時記憶装置215b内に記憶される必要のある行を減少するために、CPU201は、ステップ504での一次元IDCTを、1個の8×8画素ブロックの列順と、次の8×8画素ブロックの行順との間で交互に実行する。同様に、ステップ506での第2のバ
35 一次元IDCT動作も、行順と列順との間を交互に実行される。更に、与えられた8×8画素ブロックに対して、ステップ504で一次元IDCTが実行される順序は、ステップ506で一次元IDCTが実行される順序とは逆になっている。

【0036】本実施例では、逆量子化、コサイン乗算、及びIDCT動作は、CPU201の同じ乗算減算装置によって実行される。上述したように、本実施例の乗算命令もまたバタフライ動作を実行する。本実施例は、図8の回路600を用いた乗算動作とバタフライ動作との同時の動作を達成する。図7の回路600は、乗算減算装置601及びバタフライ装置602を有する。図7に示すように逆量子化命令の間、逆量子化乗数は量子化メモリ215aから検索され、各々は乗算器603の5ビットスケール因子によってスケールされる。スケールされた逆量子化乗数は、次に、ジグザグメモリ212から検索されたDCT係数によって乗算されるべく、マルチプレクサ604を通して、乗算減算装置601に提供される。マルチプレクサ604は、逆量子化命令の間は、逆量子化乗数を選択するべくセットされ、コサイン乗算動作の間はコサイン因子を選択するべくセットされる。コサイン因子はコサインメモリ215cから検索される。この実施例では、コサインメモリ215cは、読
40

み出し専用メモリとして実施されている。

【0037】乗算減算装置601に到達する前に、回路600の最初のステージ(図8)では、各DCT係数は、MPEG規格の最初の要求に基づいて、減数され(ブロック654)、奇数にされるかまたは0に向かって四捨五入されるか(ブロック656)、または予め決められた範囲にクリップされる(ブロック658)。

“ゲート”651、マルチプレクサ652、デクリメンタ653、ラウンダ656、及びクランプ658は、図8に更に詳しく示されている。

【0038】図8に示されているように、ジグザグメモリ212からの9ビットDCT係数は、制御信号“coded”に応答して、“ゲート”651によって0にセットされることが可能である。この9ビットデータは、14ビットデータを形成するために、右側に0のビットをつけ加えた後に、14ビットバス681上の逆量子化命令を実行する間、マルチプレクサ652によって選択される。代わりに、逆量子化命令以外の命令を実行するときには、マルチプレクサ652はバス682の14ビットデータを選択する。逆量子化命令以外の命令を実行する間のこの14ビットデータは、レジスタファイル内のレジスタ(“ソースレジスタ”)から検索されたデータの下位14ビットである。バス681上の14ビットデータは、MPEG規格によって要求されたとき、デクリメンタ653の出力端子に14ビット出力を提供するべく、デクリメンタ653によって減少させられる。もし減少動作が要求されなかったならば、バス681上の14ビットデータは、変更なしに、デクリメンタ653の出力端子に提供される。

【0039】デクリメンタ653の出力データのビット0(LSB)とビット4の両方は、バス683上に14ビットデータを提供するべく、MPEG規格に基づいて置換される。逆量子化命令の実行中に、もしジグザグメモリ212からのDCT係数が0ならば、若しくは、逆量子化命令以外の命令(例えばコサイン乗算命令)の実行中に、レジスタファイルからのデータが0ならば、バス683上の14ビットデータは“ゲート”656bによって0にされることになる。逆量子化命令の実行中には、ソースレジスタのビット23:19は、“ゲート”656bの14ビット出力データの前に置かれ、結果として得られた19ビットデータは、クランプ658によって、-2047から2047までの値を有する14ビットデータにクランプされる。一方、逆量子化命令の間には、“ゲート”656bの14ビット出力データは、クランプ658の出力データとして通過する。次にクランプ658の出力データは、バス684上の23ビットデータを形成するべく、右側に0のビットを加えられる。実行されている命令が、“imac”命令でない限りは、マルチプレクサ659は、この23ビットデータをレジスタ660の入力端子へ選択する(以下を参照の

こと)。“imac”命令の実行中、レジスタ660は、ソースレジスタからの23ビットの最下位ビットをラッチする。レジスタ660の出力データは、乗算減算装置601への“X”入力データとして提供される。

【0040】図7を再び参照すると、乗算減算装置601は、実行された命令によって(例えば、逆量子化、コサイン、及び整数乗算命令の何れかの場合)、2つの数XとYとを乗算し、(例えば、IDCT乗算減算命令の場合)、 $X \times Y - Z$ の値を計算する。DCT係数は、ジグザグメモリ212または一時記憶装置215bの何れかから、レジスタファイルへフェッチされる。更に、乗算減算装置601での動作の結果として得られた数値は、オペランドとして、レジスタファイルをバイパスしてバタフライ装置602へ直接送られることができる。

【0041】バタフライ装置602は、その2つの入力オペランドの和と差とを同時に計算する。乗算命令の実行中、乗算減算装置601及びバタフライ装置602は、各々のオペランドに対して演算を行うので、乗算命令は、乗算結果及びバタフライ結果の両方の結果を得る。更に、“パイプライン”効果は、バイパス命令に続く2番目の命令のバタフライ動作内のバイパス命令内で、乗算減算装置601の出力値(“中間”の結果)を用いることによって得られる(乗算減算装置601は、2命令サイクルの待ち時間を有する)。この構造のもとでは、中間の結果が、レジスタファイルのレジスタ内にロードされず、従ってレジスタから読み出されないの

で、高いスループットが得られる。

【0042】第1のバスIDCTのバタフライ動作の結果は、一時記憶装置215b内に送られ、一方、第2のバスIDCTのバタフライ動作の結果は、クランプ605によって“クリップ”され(clipped)、そして画素メモリ213に送られる。

【0043】図9は、図5に示されたIDCTを計算するためのマイクロプログラムを示している。図9では、逆量子化、コサイン乗算、及びIDCT動作は、図9でニモニック“dmac”、“cmac”、及び“imac”で各々示された命令によって表現されている。更に、命令“reg(a, b)”は引数“b”によって特定されたレジスタを、引数“a”で特定された名前に割り当てる。マイクロプログラムについてのコメントは、各行のシンボル“/*”と“*/”との間に提供されている。IDCT命令のコメントの部分では、バタフライ装置602及び乗算減算装置601によって実行される動作は、各々見出し“BUTTERFLYS”及び“MULTITLIES”の下に説明されている。図6では、マイクロプログラムのIDCTの部分(即ち、命令imac, dmac, 及びcmacが呼び出される部分)では、コメントの行には、1~6の番号が付されており、IDCTによって実行されるループの命令サイクルを表示している。

【0044】図9は、各々CPU201を通るデータの流れと、レジスタファイル内のレジスタの内容のマップである図3及び図4と共に参照される。図3では、行1～26は、図9に示されたIDCT部分の対応する番号を付された命令サイクルに対応する。見出し“zmem”と“tmem”の下最初の2行は、各々ジグザグメモリ212と一時記憶装置215bとからフェッチされたオペランドを示している。見出し“pmem”の下には、画素メモリ213に書き込まれたIDCTの結果の値が示されている。見出し“A”、“B”、及び“C”の下オペランドは、バタフライ装置602のX、Y入力及び乗算減算装置601のZ入力に備えられているレジスタからフェッチされたオペランドに対応する。見出し“E”の下値は、バタフライ装置602の出力から得られた結果に対応する。乗算減算装置601は、3ステージのパイプラインされた乗算減算装置からなる。即ち、見出し“E1”、“E2”、及び“E3”の下値は、乗算減算装置601の第1、第2及び第3ステージによって各々実行された動作のオペランドの内の1つに対応する。見出し“R1”、“R2”、及び“R3”の下には、レジスタファイルに書き込まれる、バタフライ装置602及び乗算減算装置601からの結果が示されている。R1及びR2は、バタフライ装置602からの和及び差の結果に対応し、R3は、乗算減算装置601の結果に対応する。

【0045】図4は、レジスタファイルのマップであって、図9に示されたマイクロプログラムのIDCT部分の命令サイクルの間、16個のレジスタR2～R17に記憶された値を示している。図4に示された行1～26の各々は、図9の対応する番号を付された命令の間のレジスタR2～R17の内容を示している。

【0046】図9では、逆量子化命令は、命令ニーモニック“dmac (BTP, r12, r, a, b)”によって表現され、

【0047】ここで、(1) BTPは、nT、rT、wT、wP、BnT、BrT、BwT、及びBwPの内の1つであって、各々は、メモリからの読み出しをしない、一時記憶装置215bからの読み出しをする、一時記憶装置215bへの書き込みをする、画素メモリ231への書き込みをする、レジスタファイルのバイパスを伴ったメモリからの読み出しをしない、レジスタファイルのバイパスを伴った一時記憶装置215bからの読み出しをする、レジスタファイルのバイパスを伴った一時記憶装置215bへの書き込みをする、及びレジスタファイルのバイパスを伴った画素メモリ213への書き込みをすることに対応する。

【0048】(2) “r12”は、バタフライ計算の結果が記憶されている2個のレジスタの内の一方のアドレスを意味する。特に、アドレスr12のレジスタは、バタフライ計算の和の部分記憶し、アドレスr12+1

のレジスタは、バタフライ計算の差の部分記憶している。

【0049】(3) “r”は、QMEM215aからの逆量子化定数を、ジグザグメモリ212の出力FIFOで、次のDCT係数に乗算する、逆量子化動作のあて先レジスタのアドレスを意味する。

【0050】(4) “a”及び“b”は、各々関連するバタフライ計算のソースレジスタを意味する。

【0051】図9では、コサイン乗算命令 (“cmac”) は、命令ニーモニック “cmac (BTP, r12, r, a, b, c)” によって表現されている。“cmac”命令の引数は、“dmac”命令に関して上述された引数と概ね等しい。コサイン乗算命令を実行するときには、DCT係数の8×8ブロック内の位置によって決定されるコサイン因子は、レジスタファイルの特定されたレジスタ“c”の内容を乗算される。

【0052】図6では、IDCT命令は、命令ニーモニック “imac (BTP, r12, r, a, b, c)” で表現されている。“imac”命令の引数は、imac命令を実行するとき、コサイン因子が引数“c”によって特定されるレジスタの内容を乗算されること以外は、概ね“cmac”命令に関して説明された引数と等しい。出力される前に、結果の積は、次の命令の引数“b”によって特定されるレジスタの内容を減算される。

【0053】特に、Xiは、ジグザグメモリ212から受け取られた量子化されたDCT係数を意味し、dXiは、量子化された後のDCT係数Xiの値を意味し、CXiは、逆量子化と、コサイン因子による乗算との両方を行われた後のDCT係数Xiの値を意味する。(コサイン因子による乗算は、図5のIDCTアルゴリズムの最初のステップとして示されている。)

【0054】図9では、添字“p”を有する名前即ち、同図に示されたマイクロコードのIDCT部分の3行目の“A_p”は、二次元IDCTアルゴリズムの第2パス内の値を意味する。一方、添字“p”を有さない名前は、二次元IDCTの第1パス内の値を意味する。IDCTの第1及び第2パスの結果は、各々一時記憶装置215b及び画素メモリ213を通過する。

【0055】バタフライ動作の和及び差の結果に割り当てられた変数名は、各々名称“0”及び“1”を付加されている。例えば、図9に示されたアルゴリズムのIDCT部分の1行目では、コメントの表示“B_p=B(X_{3p}, X_{5p})”は、“dmac”動作のバタフライ部分が、オペランドX_{3p}及びX_{5p}を取り上げ、これらのオペランドの和と差とを各々B_{0p}及びB_{1p}として計算することを表している。

【0056】乗算減算装置601の減算入力の入力データとして、及びバタフライ装置602への入力として用いられる値は、図9のマイクロプログラムのIDCT

動作のコメント部分内の“%”記号によって表示されている。例えば、IDCT部分の11行目では、表示“iB1p=imac(B1p,%B0p)”は、IMAC命令の乗算減算部分のオペランドB0pが次の命令内で用いられることを表している。即ち、12行目では、B0pが、12行目のIMAC命令のバタフライ部分のオペランドとして用いられていることを示す表示“AAp=b(A0p,%B0p)”が示されている。最後に、図9では、レジスタファイルをバイパスし、直接バタフライ装置602に向かって通過する乗算減算装置601の結果が、“\$”記号によって表示されている。例えば、4行目では、表示“\$cX4=cmac(dX4)”は、オペランドdX4に対して実行されたコサイン乗算の結果cX4が、レジスタファイルをバイパスし、バタフライ装置602に向かって直接通過することを表示している。

【0057】命令メモリ

CPUの命令を実行するために用いられるマイクロコードを記憶する命令メモリ216は、スタティック・ラム(SRAM)からなる。命令メモリ216の内容は、CPU201の初期化時に、ホスト計算機202によってロードされる。必要に応じて、マイクロコードの変更を実行するために、SRAM内のマイクロコードが、DRAM217からのマイクロコードによってオーバーレイされる。

【0058】メモリコントローラ

DRAMコントローラ206は、DRAM217と直接対話し、外部DRAM217への書き込み及び外部DRAM217からの読み出しに必要な信号を発生する。DRAMコントローラ206は、命令FIFO207を通して、CPUから開始アドレス及びオフセット情報を受け取る。DRAMコントローラ206は、複数の転送がリクエストされたとき、次のアドレスを計算する。このため、CPU201は、各メモリアクセス毎に各アドレスを発生する必要がないので、CPU201は、IDCT動作のためのより広い帯域幅を備えている。

【0059】図10は、本実施例のメモリコントローラモジュールのブロック図である。図10に示すように、メモリコントローラモジュールは、DRAMコントローラ206及び転送リクエストFIFO(TRF)207としても知られている命令FIFO207を有する。未決定のDRAMアクセスは、(1)バイトDRAMアクセスの開始アドレスと、(2)リクエストされたアクセスが読み出しアクセスか、書き込みアクセスかの何れかであることと、(3)フェッチされるべきメモリワードの数(“長さ”)と、(4)もし適切ならば、オフセットとを表示しているエントリをCPU201がメモリバッファ108へ書き込むことによって、初期化される。本実施例では、メモリバッファ101は、データソースまたはあて先に対する優先順位の順番に配置された、

(1)輝度ビデオFIFO208a用の1つのエントリと、(2)コードFIFO204用の1つのエントリと、(3)デコーダFIFO210用の1つのエントリと、(4)画素メモリFIFO213用の5個のエントリと、(5)ホストメモリリクエスト用またはCPUメモリリクエスト用の1つのエントリと、(6)色光度ビデオFIFO208b用の1つのエントリとからなる、11個のDRAMアクセスリクエストエントリを保持している。メモリコントローラ206の動作を理解するために、各ソースまたはあて先に配置された1個または複数のエントリを1個のFIFOであると考えることができる。

【0060】DRAMアクセスリクエストは、(1)CPU201が、メモリバッファ801内にラッチされていたメモリアクセスリクエストエントリをレジスタ804内に書き込んだ後と、(2)データソースまたはあて先に対応するリクエストラインに信号が発生した後とに、未決定となる。状態レジスタファイル803a~803fは、各リクエストラインに対して、メモリアクセスリクエストが未決定であるかどうかを表示するレジスタを提供する。本実施例は、画素メモリ213に対して5個のエントリを配置しているので、画素メモリ213に関連する未決定のメモリアクセスリクエストの数を表示するために、5個のビットが提供されている。もちろん、残りのデータソースまたはあて先に対応する他の状態レジスタの各々に、1ビットが提供される。読み出しリクエスト信号を発生させることにより、メモリバッファ801内のエントリは、CPU201によって読み出すことができる。そのエントリは、CPU201によって、レジスタ805から読み出される。

【0061】各DRAMアクセスが終了したとき、TRFエントリの長さフィールドの下位3ビットが0の時以外は、フェッチさせるべきメモリワードの長さが1だけ減算される。TRFエントリの長さフィールドの値が0であることは、各メモリアクセスの後に、特定のオフセットが長さから減算されるべきであることを表示している。TRFエントリ内の各エントリもまた、転送のタイプと、バイト書き込み(即ち、16ビットのメモリワードの内の8ビットだけが重ね書きされる)が実行されるかどうかとを表示する。

【0062】本実施例では、ホスト計算機202が、TRF207内への書き込みをするのみで、TRF207からの読み出しをしないにもかかわらず、ホスト計算機202もまた、CPU201のリクエストラインではなく、CPU201が用いるのと概ね等しい方法で、ホストリクエストラインを用いて、DRAMアクセスをリクエストする。

【0063】未決定のメモリアクセス間の優先順位の裁定は、上述の優先順位方式に基づいて、優先順位裁定回路802によって行われる。DRAMコントローラ20

6が使用されていないならば、TRF207によって、最も高い優先順位のリクエストが、DRAMコントローラ206に送られて、レジスタ806内に書き込まれる。DRAM206が優先順位の低いリクエストを処理しているときに、優先順位の高いリクエストが、TRF207によって受け取られたならば、優先順位裁定回路802は、DRAMコントローラ206に、“より優先順位の高いリクエストが未決定である”という信号を送る。この実施例では、より優勢順位の高いリクエストが未決定中に、現在のメモリがページの境界にまたがってアクセスしているならば、DRAMコントローラ206は、未完成の優先順位の低いリクエストをTRF207に戻し、より優先順位の高いリクエストの処理を開始する。

【0064】DRAMコントローラ206が使用されていないで、かつDRAMアクセスリクエストがTRF207で未決定でないとき、TRF207は、コードFIFO204の内容をDRAM217に転送するための“コードFIFOエンプティ”リクエストを発生させる。コードFIFO208のメモリアksesリクエストラインに信号が発生していなくとも、コードFIFO204のリクエストに対応する有効なTRFエントリが存在し、かつコードFIFO204が少なくとも1個または複数のワードを含んでいる間は、このコードFIFOエンプティリクエストは発生している。コードFIFOエンプティリクエストは、コードFIFO204の最後の数ワードがDRAM217へ転送されることを確実にする。

【0065】DRAMコントローラ206は、TRF207によって書き込まれた、DRAMアクセスリクエストをレジスタ806から受け取る。レジスタ806内のDRAMアクセスリクエストのフォーマットは、メモリバッファ801内のTRFエントリのフォーマットと等しい。アドレス発生ロジック807は、受け取ったDRAMアクセスリクエストの開始アドレス、長さ、及びオフセット情報に基づいて一連のメモリアドレスを計算する。DRAMコントローラ206は、ステートマシン810によって制御され、そのステートマシンは、現在のDRAMがページの境界にまたがってアクセスしているときに、上述されたように、他の未決定のDRAMアクセスリクエストと、未完成のDRAMアクセスリクエストの優先権を検討し、かつ処理する。

【0066】DRAMコントローラ206がDRAMアクセスを終了したとき、TRF207が、終了したDRAMアクセスと等しいソースまたはアて先からの新しいリクエストに、メモリバッファ801内のTRFエントリを配置するように、“メモリリクエストダン”信号がTRF207に送られる。この実施例では、DRAMコントローラ206は、“オールモストダン”信号を次のときに送る。(1)現在のDRAMアクセスの終了の数

サイクル前。(2)現在のアクセスを打ち切る“キル”信号がTRF207から受け取られた時。(3)現在のDRAMアクセス中にページクロスが予想され、かつより優先順位の高いDRAMアクセスリクエストが未決定の時。“オールモストダン”信号が発生したとき、バス812をTRF207とDRAMコントローラ206との間のコミュニケーションのために開放するため、TRF207へのCPU201のアクセスがディスエーブルされる。

【0067】DRAMコントローラ206は、必要なインタフェース信号をDRAM217へ提供し、かつDRAM217のリフレッシュアクティビティを制御する。リフレッシュカウンタは、リフレッシュが実行される前に、サイクル数のトラックを保存する。もし、リフレッシュカウンタのカウンタが0になる前に、DRAMコントローラ206が使用されていない状態になったならば、DRAMリフレッシュが実行される。一方、リフレッシュカウンタのカウンタが0になったならば、DRAMリフレッシュは、現在のDRAMアクセスの終了後に、またはページ境界がクロスされたときに実行される。

【0068】可変長コード(VLC)デコーダ

DRAMコントローラ206及び画素フィルタ214と同じように、VLCデコーダ211は、CPU201に対するスレーブプロセッサとして働く。VLCレコーダ211の命令は、次の機能を実行する。(1)DRAM217から検索された可変長コードのストリームを、CPU201の指図のもとにあるデコーダFIFO210内に受け取る。(2)MPEG規格に基づいて、可変長コードをデコードする。(3)CPU201内での“非ジグザグ”化及び逆量子化のために画素の8×8ブロックを構成する。(4)同時に、コードストリームのビットの内の15個以上のビットを提供する。

【0069】図13は、VLCデコーダ211と、デコーダFIFO210とを含むVLCデコーダモジュールを示す。図13に示すように、デコーダFIFO210は、グローバルバス209を通してDRAM217から可変長コードのストリームを受け取る。制御情報(即ち、命令)もまた、CPU201から受け取られ、そしてグローバルデータデコード装置1106の一部であるデコーダ命令レジスタ内に記憶される。ある可変長コードのデコードされた値が、9ビットデータバス1101上のジグザグメモリ212に提供される。VLCデコーダ211の他の出力値は、グローバルバス209上に提供される。ステータスレジスタ(図示されていない)は、グローバルバス209を通してCPU201によってアクセスされるステータス情報を提供する。

【0070】VLCデコーダ211への命令は6ビット幅である。セットの時、ビット5(即ち、最上位ビット)は、VLCデコーダ211をリセットする。通常の

動作中即ちビット5が0のときには、下位の5ビット(4:0)は、(1)コードストリームからグローバルバス209へのビット1~15を出力する、15個の“ゲットビット”命令の内の1つ、及び(2)残りのVLCデコーダ命令、の内の何れかをエンコードする。これらの残りのデコーダ命令は、“mba”(マクロブロックアドレス)、“mtyp ei”(内部フレームマクロブロック)、“mtyp ep”(予測フレームマクロブロック)、“mtyp eh”(h. 264タイプマクロブロック)、“mtyp eb”(双方向性マクロブロック)、“mv”(動作ベクトル)、“cbp”(デコードされたブロックパターン)、“luma”(輝度ブロック)、“chroma”(色光度ブロック)、及び“non-intra”(DC成分を伴わないブロック)からなる。ここで、“mtyp eh”以外のこれらの命令は、MPEG規格で定義されたデータオブジェクトに対応する。“mtyp eh”は、テレビ会議用アプリケーションで用いられるh. 264規格によって定義されたマクロブロックを表している。これらの残りのVLCデコーダ命令は、VLCデコーダ211による、コードストリームの“ヘッド”での可変長コードのデコードを指図する。そのデコードされた値が、Zデータバス1101上の出力である、“luma”、“chroma”、及び“non-intra”命令を除いて、VLCデコーダ211の命令のデコードされた値の出力は、(レジスタ1102a~1102dで形成された)デコーダレジスタ内に記憶され、グローバルバス209上のCPU201に提供される。

【0071】本実施例では、デコーダFIFO210は、FIFOアドレスロジック装置1103内に保持されている5ビットの書き込み及び読み出しポインタによってアドレス指定可能な、32×16ビットFIFOからなる。フリーズロジック装置1104は、デコーダFIFO210が空のとき、VLCデコーダ211の動作を停止する。VLCデコーダ211は、図13に1024×15ビット読み出し専用メモリ(ROM)1105として示されている制御記憶装置によって制御され、ROM1105はまた、可変長コードのデコードされた値を記憶する。ROM1105内の可変長コードのデコーディングは、表検索によって実行される。

【0072】図13に示された本実施例では、デコーダ命令レジスタの命令が、“ゲットビット”命令(即ち、命令のビット4が0)のとき、ROMアドレス発生器1107は、(1)予め決められたビットパターン(この場合、6ビット値101011)と、(2)その命令の下位4ビットとからなるアドレスを発生させる。命令が、“ゲットビット”命令以外の命令ならば、(1)先頭の値0の2個のビットと、(2)その命令の下位4ビットと、(3)コードストリームのヘッドの最初の4個のビットとからなる10ビットアドレスを発生させる。

【0073】VLCデコーダ211によってデコードされた可変長コードが12ビットになる可能性があるので、発生させられたROMアドレスからわかるように、コードストリームの内の高々4ビットが、ROM1105へのアクセス毎に用いられ、与えられた可変長コードのデコーディングには、複数のクロックとROM1105への複数のアクセスが必要となる。“luma”のような他の命令もまた、終了するためには、複数のクロックとROM1105への複数のアクセスを必要とする。ROM1105の現在のワード“現在のROMワード”の最上位ビット(14)がセットされたとき、現在の命令の実行が終了したこと、または、現在の命令がブロック命令(即ち、“luma”、“chroma”、及び“non-intra”の内の何れか)であるときは、ラン長さ(runlength)が特定されたことを表示する。ラン長さが特定されたとき、(特定されたラン長さと等しい)0の数が、Zデータバス205へ出力されるために“アンパック”される(unpacked)。ブロック命令は、次に説明するような特別な取扱いを必要とする。

【0074】現在ワードのビット13及び12は、コードストリームのヘッドを進めるために、ビットの数をデコードする。本実施例では、コードストリームのヘッドを進めることは、ビットストリームロジック1108の制御のもとで、左シフト1109及び右シフト1110によって実行される。

【0075】現在のROMワードのビット14が、現在の命令の実行が終了していないことを表示する値0であるとき、次のROMアドレスを提供するために、現在のROMワードの6個のビット(9:4)は、(1)現在のROMワードのビット11がセットされているときは、コードストリームのヘッドの次の4個のビットと、(2)現在のROMワードのビット10がセットされているときは、現在のROMワードの他の4個のビット(3:0)と結合される。現在のROMワードのビット11の値は、次のROMアクセスがデコーディングの目的のためであり、従って次のROMアドレスの残りの4個のビットがコードストリームのヘッドから取り出されることが必要であることを表示する。一方、次のROMアクセスが、現在のROMワードのビット10の値によって表示されるように、制御の目的のためであるならば、次のROMアドレスの残りの4個のビットは、現在のROMワードのビット(3:0)から取り出される。

【0076】現在の命令がブロック命令であって、デコードされたAC値—ラン長さの組のラン長さの部分が特定されたことを表示するべく、現在のROMワードのビット14がセットされているならば、次のROMアドレスは、予め決められた4ビットパターン(本実施例では、1101)及び特定された6ビットのラン長さによって形成される。特定されたラン長さは、(1)ブロックの終わり(EOB)のシンボルが特定されたときは、

z データカウンタ 1111 の中に、(2) 短いエスケープシンボルが特定されたときは、レジスタ 1102b とレジスタ 1102c との内容をカスケードにすることによって得られた値の中に、(3) 長いエスケープシンボルが特定されたときは、レジスタ 1102a とレジスタ 1102b との内容をカスケードすることによって得られた値の中に、(4) それ以外の時は、現在の ROM ワードのビット (10:6) の中に見いだせる。次のような 3 つの違法な範囲がある。(1) ビットパターン 1000_0000_0000_0000 によって表現される値。(2) 1000_0000_1000_0001 から 1000_0000_1111_1111 までの範囲。(3) 0000_0000_0000_0000 から 0000_0000_0111_1111 までの範囲。レベルコードが正当な範囲内にあることの確認は、デコーディング動作に関する前述された方法と同じように、最後にアクセスされた ROM アドレスの特定のビットを用いて、クロック周期毎に、コードストリームからシフトされた 4 個のビットを ROM の下位アドレスへ写すことによって完了する。もし 16 ビットのレベルコードが違法な範囲内にあるならば、到着した ROM 内のアドレスの内容は、違法な 16 ビットレベルコードの信号を送る。

【0077】本実施例では、デコーダ FIFO 210 出力レジスタ、16 ビットレジスタ 1112、5 ビットレジスタ 1113、4 ビットレジスタ 1102d、4 ビットレジスタ 1102c、4 ビットレジスタ 1102b、及び 3 ビットレジスタ 1102a が、7 ステージのパイプラインされたデータバスを形成する。更に、レジスタ 1102a ~ 1102d もまた、15 ビットレジスタとして取り扱われることが可能である。

【0078】各クロック周期の始めに、レフトシフタ 1109 は、レジスタ 1113 にコードストリームのヘッダの 5 個のビットを提供する。その内の 4 個のビットは次の ROM ワードをアクセスするために用いられ、その 4 個のビットは、次のクロック周期にコードストリームのヘッダを進めるためのビットの数 (4 ビット以上) を提供する。本実施例では、コードストリームのヘッダは、ビットストリームロジック装置 1108 内のビットポインタによってモニタされる。ビットポインタが、レジスタ 1112 のビット 0 でのコードビットを通過して、(最下位ビットに向かって) 進んだ後の 1 クロック周期では、デコーダ FIFO 210 の出力レジスタの内容が、レジスタ 1112 内にロードされ、デコーダ FIFO 210 の次のエントリが、デコーダ FIFO 210 の出力レジスタにロードされる。デコーダ FIFO 210 の出力レジスタの上位 9 ビットは、レフトシフタ 1109 が利用可能なので、(デコーダ FIFO 210 の出力レジスタ内の) コードストリームのヘッダの 5 個のビットは、立ち往生することなしに、次の ROM アドレス

を形成するべく、レフトシフタ 1109 によって提供される。4 個のビットのみが次の ROM アドレスを形成するために使用され、コードストリームのヘッダの 5 番目のビットは、後続の振幅値の符号を決定するべく、AC 係数を決定した後に、ブロック命令内で直ちに使用される。

【0079】レフトシフタ 1109 の 5 ビット出力の右シフトを提供することに加えて、ライトシフタ 1110 もまた、“luma” ブロック命令及び “chroma” ブロック命令の DC 及び AC 成分のシフトされた値に符号ビットをつける。上述されたように、VLC デコーダ 211 の制御は、ROM 1105 によって実行される。例えば、AC 係数—ラン長さの組のラン長さのデコーディングの後に、現在のブロック命令の実行の終わりまでアクセスされた各 ROM ワードは、z データカウンタ 1111 を直接減少させ、そして z データバス 205 上に値 0 が出力されることを可能にする。

【0080】ライトシフタ 1110 及びレフトシフタ 1109 は、“ゲットビット” 命令のビットのシフトを提供する。1 クロック周期当たり多くとも 4 ビットがシフトされるので、4 ビットよりも多くのビットを得るためには、複数のクロック周期が必要となる。“ゲット n ビット” 命令のための、最初のクロック周期では、(4 を法とする) n ビットは、右シフトされ、残りの数のビットは、レジスタ 1102a ~ 1102d によって形成されたパイプライン内へ、同時に連続して 4 ビット分シフトされる。

【0081】VLC デコーダ 211 の出力値が、現在の ROM ワードから取り出されたとき、グローバルバス 209 のビット 14 ~ 10 は、0 にセットされ、現在の ROM ワード 9 ~ 0 は、マルチプレクサ 1114a 及び 1114b を通してグローバルバス 209 上のビット 9 ~ 0 として出力される。もし VLC デコーダ 211 の出力値がコードストリームから取り出されるならば、マルチプレクサ 1114a 及び 1114b は、各々レジスタ 1102a 及びライトシフタ 1110 の出力データを選択する。マルチプレクサ 1114a 及び 1114b は、各々反転出力値を提供するために選択される。そのような反転出力値は、ブロック命令の実行中に、もし必要であれば、DC 振幅値の 1 の補数または AC 振幅値の 2 の補数を提供するために望ましい。z データインクリメント 1113 は、1 の補数計算または 2 の補数計算を完了する。

【0082】画素フィルタ及び動作補償

画素フィルタ 214 は、メモリコントローラ 206 からの基準フレームを受け取り、かつ CPU 201 からの復元されたビデオデータを、画素メモリ 213 から検索する。MPEG 規格に基づけば、基準フレームは、ビデオデータを基準フレームに関連づける (“予測する”) 1 個または複数のモーションベクトルを用いて、復元され

たビデオデータと結合される。結果として得られた映像は、後の出力のために、ビデオFIFO217及びビデオバス205によってDRAM217に記憶される。MPEG規格のもとでは、復元されたビデオデータは、予告を表現せず（即ち、基準フレームとは無関係であり）、逆の予告を表現し（即ち、過去の時刻の基準フレームに依存する）、先の予告を表現し（即ち、将来の時刻の基準フレームに依存して）、または、内捜された予告を表現（即ち、将来の時刻の基準フレーム及び過去の時刻の基準フレームの両方に依存）する。

【0083】本実施例では、ビデオデータが“否予測”タイプでなければ、1個または複数の基準フレームのブロックが、DRAM217からフェッチされる。これらのブロックは各々、 9×9 要素からなる。DRAM217の各ページは、8列及び32行の画素（1ページ当たり256個の画素）を記憶するので、 9×9 のフェッチは、少なくとも1つのページ境界と交差する。（実際、DRAM217の1ワードに2個の画素が記憶されているので、実際のフェッチは、 10×9 ブロックの画素を含む。）ページとの交差を最小にするために、本実施例では、次のページメモリ内に存在するブロックの残りの画素がアクセスされる前に、あるメモリページ内に存在する 9×9 ブロックの画素の全てがアクセスされる、アクセス方法を用いている。この方法は、米国特許出願第07/669,818号明細書に開示された実施例の動作補償に関連して説明されている。この特許出願明細書は、ここで言及したことによって本出願の一部とされたい。

【0084】図11は、画素フィルタ213のブロック図である。画素ペアはDRAM217からフェッチされ、かつグローバルバス209上に画素フィルタ213が備えられている。動作ベクトルは、各々XレジスタとYレジスタ（図示されていない）とに記憶されたX成分とY成分とからなる。動作ベクトルのX成分は、フェッチされた画素の 10×9 ブロック内の最初の行が 9×9 画素基準ブロックの一部であるかどうかを表示する。動作ベクトルのY成分は、フェッチされた 10×9 ブロックの幾つの列が最初のメモリページ内に存在するかを表示する。

【0085】他のサイクル毎に、画素ペアはグローバルバス209に到着し、各サイクル毎に、画素フィルタ213は1画素を処理する。 9×8 ビットのランダムアクセスメモリからなる行メモリ901は、以前にアクセスされた最後の行の画素を記憶する。現在の行の画素が到着したとき、到着した各々の画素は、加算器902によって、行メモリ901に記憶された同じ列の画素と平均化される（X方向にフィルタされる）。次に到着した画素は、行メモリ901に記憶された対応する画素に置き換えられる。加算器902の結果は、パイプラインレジスタ903内にラッチされる。

【0086】次に、フィルタされた画素は、加算器905によって、列メモリ904に記憶されていた先行する列のフィルタされた画素と平均化（y方向にフィルタ）される。パイプラインレジスタ903から入ってくる各画素は、列メモリ904内の対応する画素と置き代わる。加算器905からの結果として得られたフィルタされた画素は、パイプラインレジスタ906内に連続的にラッチされる。x方向及びy方向での平均の最終結果は、MPEG規格によって要求されるような、 $1/2$ の画素による 8×8 ブロックの変換（“再サンプリング”）である。次に、再サンプリングされた基準フレームは、加算器906によって、画素メモリ213内の復元されたビデオデータに加えられる。画素213は2つの部分からなり、各部分は、交互に、二重のバッファ方式内にCPU201からの復元されたデータを受け取り、かつ画素フィルタ213内の画素フィルタリングへ画素を提供する。

【0087】本実施例では、xレジスタ及びyレジスタが、前方動作ベクトル及び後方動作ベクトルのために備えられている。内捜された予測ブロックを処理するとき、（前方動作ベクトルに関連した）前方基準フレームは、前方補償のために最初にフェッチされる。前方補償の結果は、前方補償の後にフェッチされる後方基準フレームを用いた後方補償のために、画素メモリ213に記憶される。

【0088】ビデオインタフェース

フィルタされ、そして動作補償されたビデオデータは、ビデオデコーダ200の出力として、ビデオインタフェースへのビデオバス205上に提供される。ビデオインタフェースのブロック図が、図12に示されている。図12に示すように、ビデオデータは、画素の組として、グローバルバス209を通して、（図12では符号1000によって概ね表示されている）ビデオインタフェースに提供される。CPU201はまた、グローバルバス209を通してビデオインタフェース1000に制御情報を提供する。そのような制御情報は、例えば、YUV表現されたデータ（即ち輝度/色光度表現）とRGB表現されたデータとの間及びスキャンライン上のアクティブデータの開始位置と終了位置との間の変換のために必要な変換因子を含む。変換因子は、レジスタ1001に記憶されている。同期化信号YSYNC及びHSYNC（垂直及び水平同期化信号）を受け取るタイミングロジック1002は、ビデオインタフェース1000の動作を受け取られたビデオデータストリームに同期させる。

【0089】入ってくる画素の各組の画素は、YUV表現されていて、かつ等しいY、U、及びV型である。これらの画素は、各々がビデオFIFO208a及びビデオFIFO208bと呼ばれる2個のFIFOからなるビデオFIFO208に記憶されている。ビデオFIFO

0208a及びビデオFIFO208bは、各々輝度(Y)データ及び色光度(UまたはV)データを記憶する。

【0090】本実施例では、YUV表現されたデータは、ユーザの選択によって、出力のために、RGB表現されたデータに変換されることもできる。YUV表現されたデータからRGB表現されたデータへの変換は、ブロック1003内で実行される。外部から提供されるビデオクロック信号VCLKを受け取る同期化回路1004は、所望の速度で、24ビットバス1006上に出力ビデオデータを提供する。

【0091】これまでの詳細な説明は、本発明の特定の実施例を例示するために提供されたものであって、本発明に制限を加えることを意図するものではない。本発明の技術的視点内で種々の変形、変更が可能である。本発明は、添付の請求項によって定義される。

【0092】

【発明の効果】本発明に基づく復元プロセッサによって、ビデオ信号のリアルタイムの復元と、現在の計算機または集積回路技術を用いた経済的な実施が可能になった。

【図面の簡単な説明】

【図1】MPEG規格での、圧縮及び復元プロセスのモデルを表す図。

【図2】本発明に基づく、ビデオ用復元プロセッサ回路200のブロック図。

【図3】CPU201のデータの流れを表す図。

【図4】IDCT動作の命令サイクル内に用いられたレジスタの内容を示す、CPU201内のレジスタファイルのマップを示す図。

【図5】一次元IDCTが、DCT係数の8×8ブロックの1列または1行を構成する8個のDCT係数上で実施される、8ポイントIDCTアルゴリズムの1つのバスの流れ図。

【図6】CPU201での、逆量子化動作及びIDCT動作の順序を示す図。

【図7】本発明に基づく、乗算とバタフライ動作とを同時に、CPU201内で達成する回路600を示す図。

【図8】CPU201のデータバス内の回路600の第1ステップを示す図。

【図9】本発明に基づく、図5のIDCTアルゴリズムを用いた、CPU201内のIDCTを計算するためのマイクロプログラム。

【図10】転送リクエストFIFO(TRF)207及びDRAMコントローラ206を示す、メモリコントローラのブロック図。

【図11】画素フィルタ、及び画素フィルタ213と画素メモリ214とを有する動作圧縮モジュールのブロック図。

【図12】ビデオ用FIFO208とYUV/RGB発

散回路を有するビデオ用インタフェースのブロック図。

【図13】VLCデコーダ211とデコーダFIFO208とを含むVLCデコーダモジュールのブロック図。

【符号の説明】

- 101 フレーム間の冗長性の除去ステップ
- 102 DCTステップ
- 103 量子化ステップ
- 104 エンコーディングステップ
- 105 デコーディングステップ
- 106 逆量子化ステップ
- 107 IDCTステップ
- 108 実際の画素の発生ステップ
- 200 ビデオデコーダ
- 201 CPU
- 202 ホスト計算機
- 203 ホストバス
- 204 コードFIFO
- 205 ビデオバス
- 206 DRAMコントローラ
- 207 命令FIFO
- 208 ビデオFIFO
- 209 グローバルバス
- 210 デコーダFIFO
- 211 VLCデコーダ
- 212 ジグザグメモリ
- 213 画素メモリ
- 214 画素フィルタ
- 215 メモリ装置
- 215a 量子化メモリ
- 215b 一時記憶装置
- 215c コサインメモリ
- 216 命令メモリ
- 501 ジグザグメモリからの検索ステップ
- 502 逆量子化ステップ
- 503 コサイン乗算ステップ
- 504 一次元IDCT1ステップ
- 505 一時記憶装置への記憶ステップ
- 506 一次元IDCT2ステップ
- 507 画素メモリへの提供ステップ
- 601 乗算減算装置
- 602 バタフライ装置
- 603 乗算器
- 604 マルチプレクサ
- 605 クランプ
- 801 メモリバッファ
- 802 優先順位裁定回路
- 803a～803f 状態レジスタ
- 804 FIレジスタ
- 805 FPレジスタ
- 806 MFレジスタ

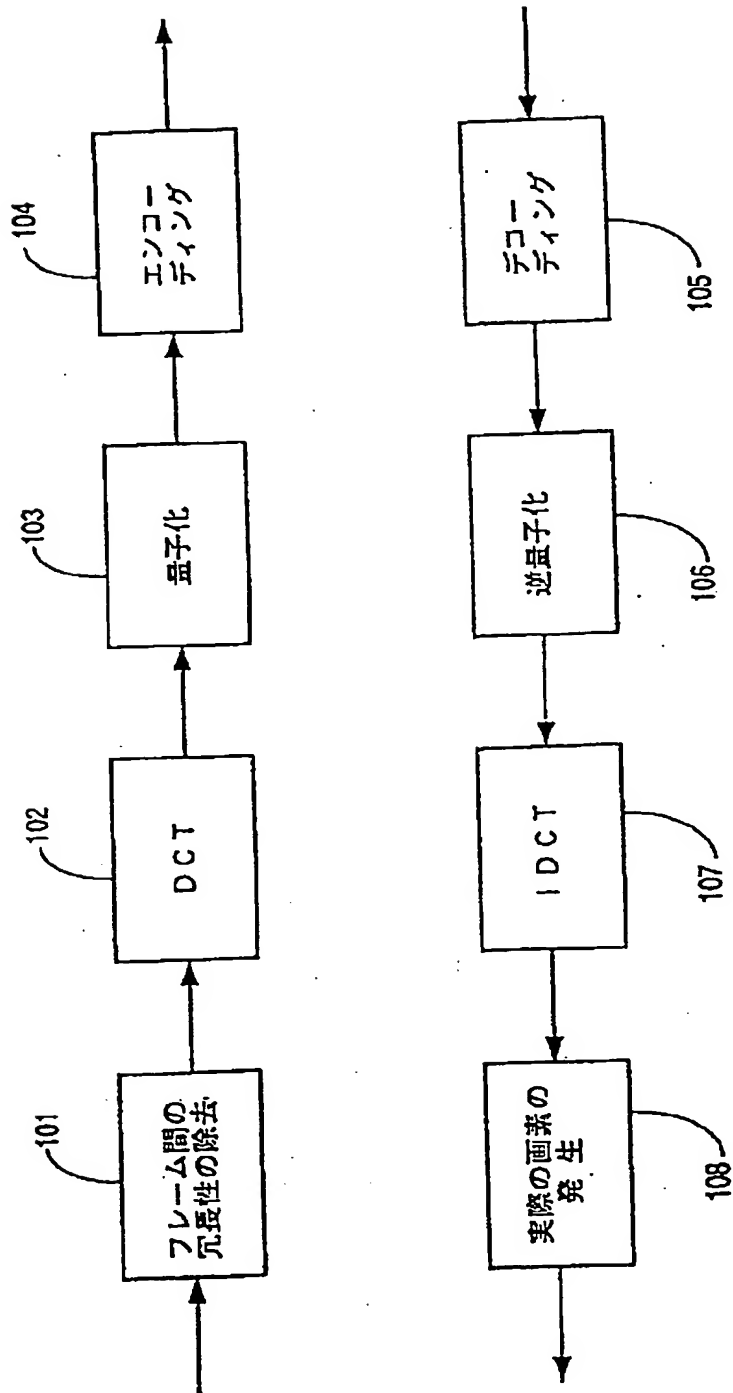
807	アドレス発生ロジック
808	MSレジスタ
901	行メモリ
902	加算器
903	パイプラインレジスタ
904	列メモリ
905	加算器
906	パイプラインレジスタ
907	加算器
1001	レジスタ
1002	タイミングロジック
1003	補間及び色変換装置
1004	同期化回路
1005	出力ラッチ
1006	24ビットバス

	1 0 0 7	境界搜入装置
	1 1 0 2 a ~ 1 1 0 2 d	レジスタ
	1 1 0 3	F I F O アドレスロジック装置
	1 1 0 4	フリーズロジック装置
	1 1 0 5	ROM
	1 1 0 6	グローバルデータデコード装置
	1 1 0 7	ROMアドレス発生器
	1 1 0 8	ビットストリームロジック
	1 1 0 9	左シフト
10	1 1 1 0	右シフト
	1 1 1 1	z データカウンタ
	1 1 1 2	レジスタ
	1 1 1 3	レジスタ
	1 1 1 4 a、1 1 1 4 b	マルチプレクサ

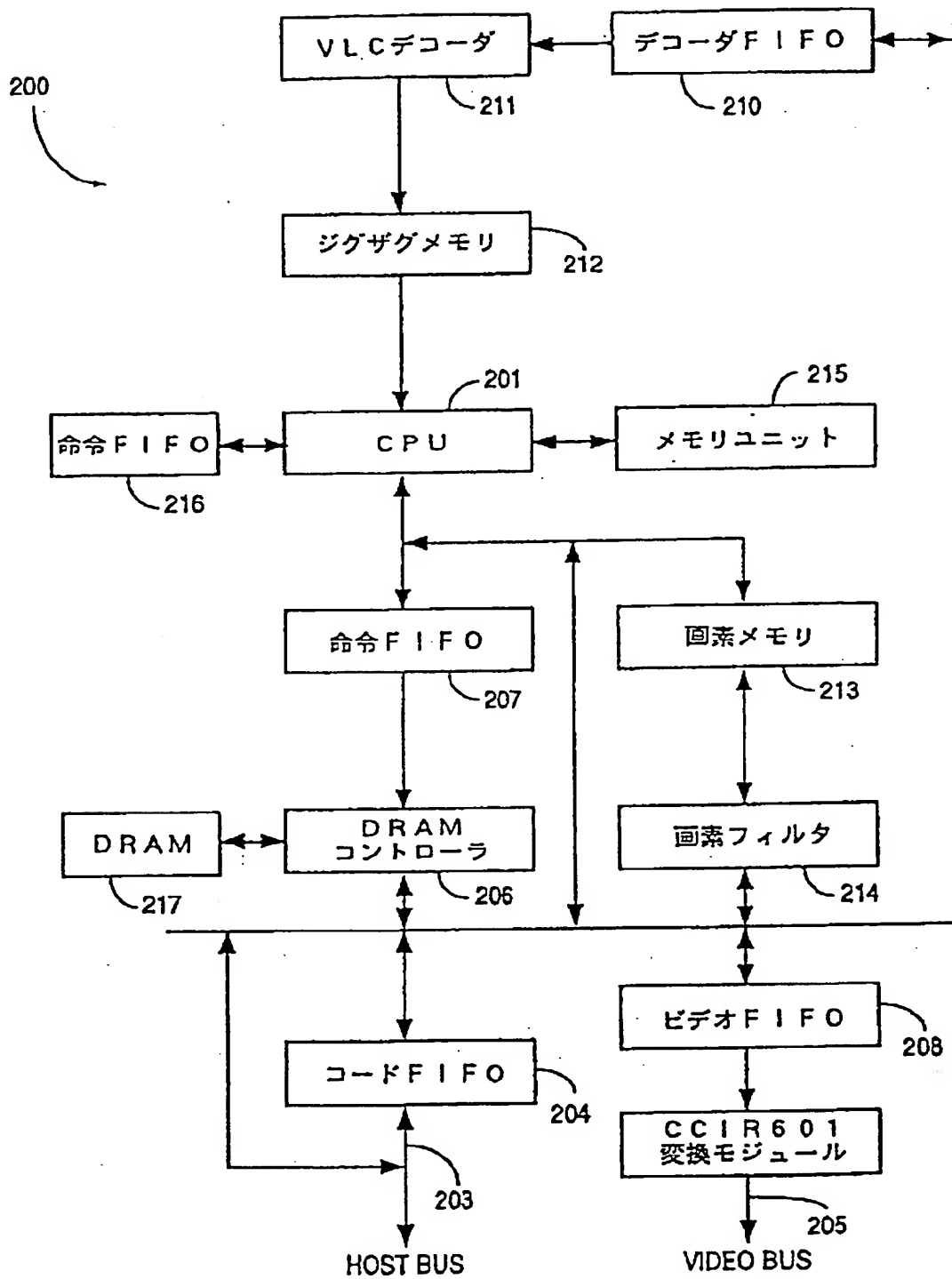
【図 3】

[illegible]

【図1】



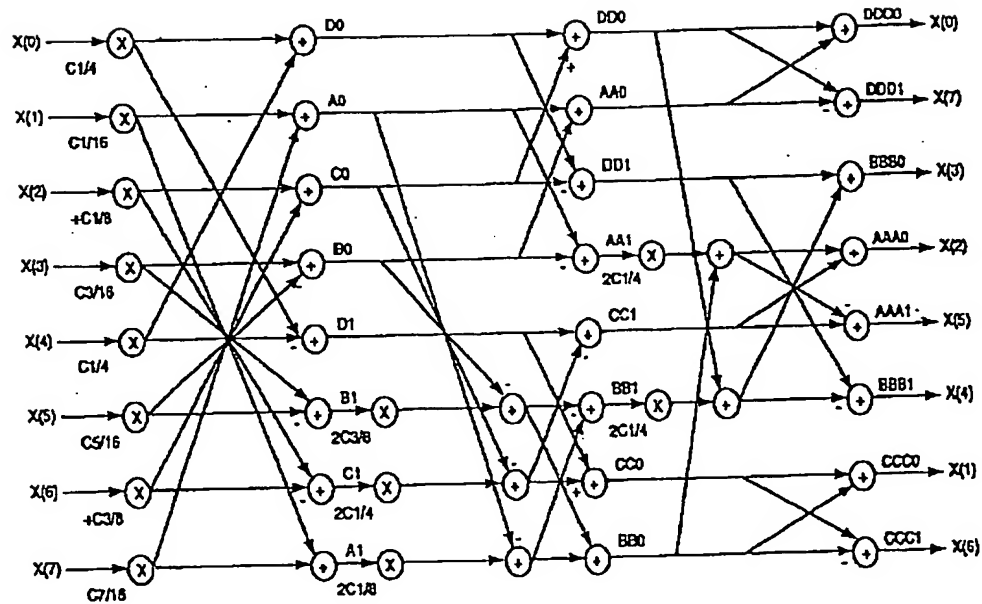
【図2】



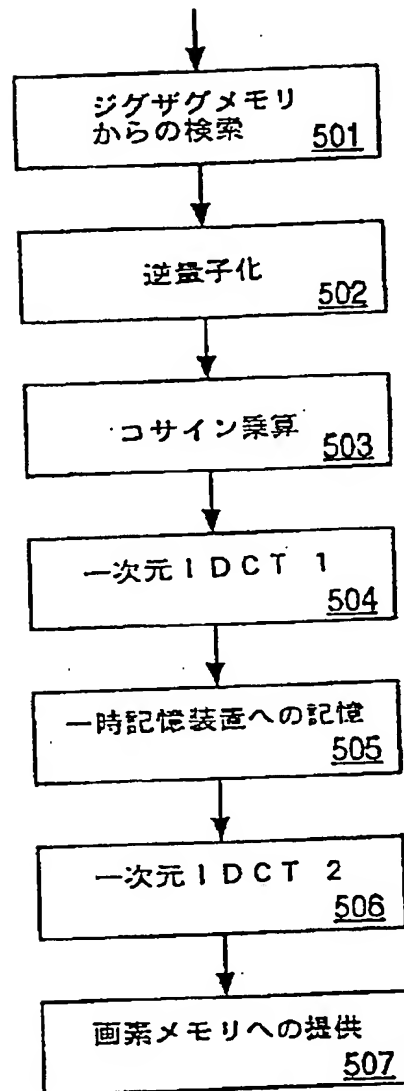
【図 4】

mpa base	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19
0	A0	A1	B0	B1	C0	C1	C2	C3	C4									
1	A0	A1	B0	B1	C0	C1	C2	C3	C4									
2		A1	B1	B1	C2	C2	C2	C2	C4									
3		A1	B1	B1	C2	C2	C2	C2	C4									
4		A1	B1	B1	C2	C2	C2	C2	C4									
5	C0	A1	B1	C0	C1	C1	C1	C1	C1	A1p	A1p	B1p	B1p					
6	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
7	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
8	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
9	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
10	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
11	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
12	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
13	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
14	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
15	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
16	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
17	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
18	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
19	C0	C0	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
20			B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
21			B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
22			B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
23	A0	A1	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
24	A0	A1	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
25	A0	A1	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
26	A0	A1	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
27	A0	A1	B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
28			B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
29			B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
30			B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			
31			B0	B1	C0	C0	C0	C0	C0	A1p	A1p	B1p	B1p	C0p	C0p			

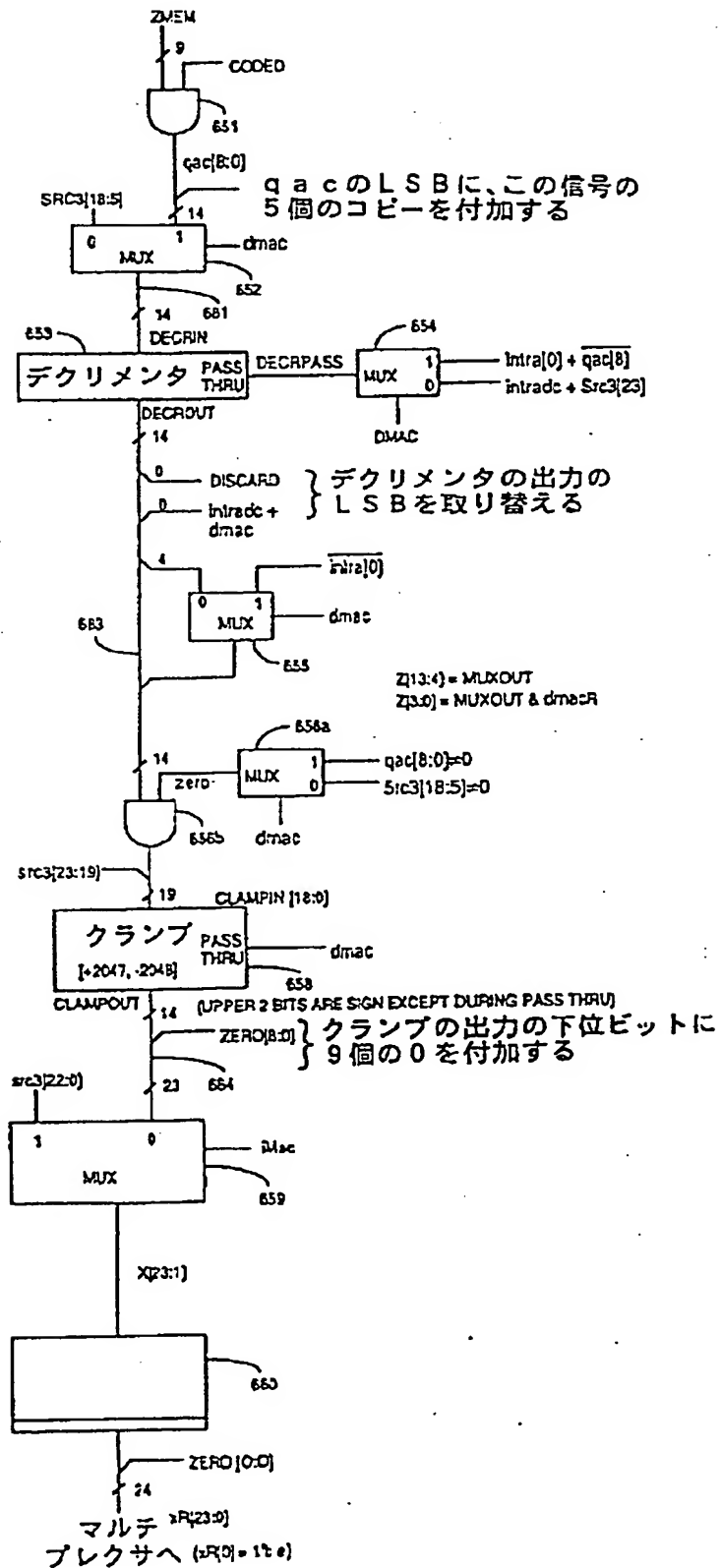
【図 5】



【図6】



【図8】



【図9】

```

ucoda () (
reg (loopcnt, r1)

/* Registers used for dequantization and cosine multiply */
reg(dx0,r17)    reg(dx1, r19)    reg(dx2, r6)    reg(dx3, r2)
reg(dx4,r9)     reg(dx5, r3)     reg(dx6, r7)    reg(dx7, r5)
reg(cx0,r8)     reg(cx00, r13)    reg(cx1, r19)   reg(cx2, r6)
reg(cx3,r2)     reg(cx6, r7)

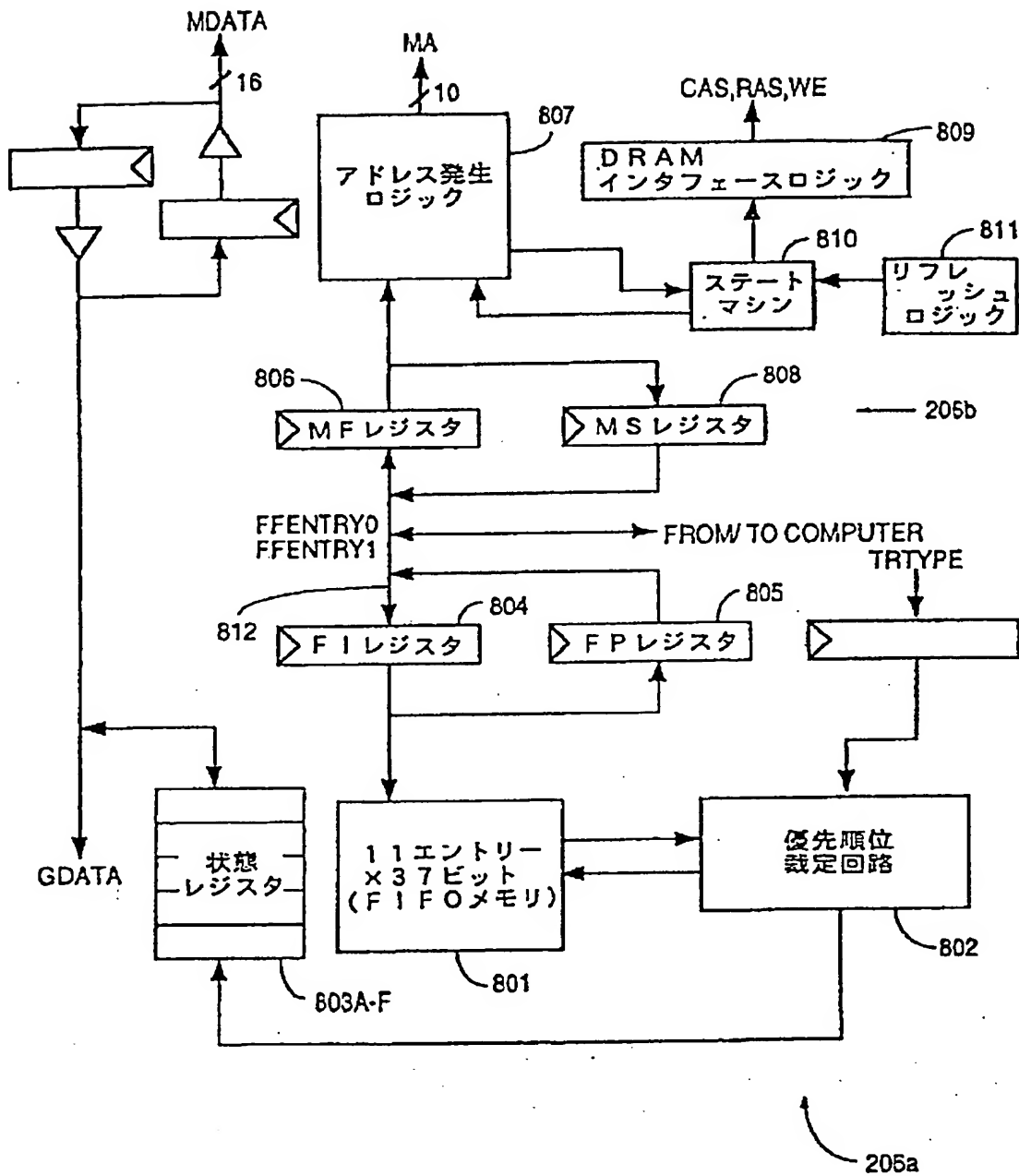
/* Registers used for 1st pass of IDCT (ZMEM to TMEM) */
reg(A,r2)       reg(A0, r2)      reg(A1,r3)      reg(iA1, r3)
reg(B,r4)       reg(B0, r4)      reg(B1,r5)      reg(iB1, r5)
reg(C,r6)       reg(C0, r6)      reg(C1,r7)
reg(D,r8)       reg(D0, r8)      reg(D1,r9)
reg(AA,r18)     reg(AA0, r18)    reg(AA1,r19)
reg(BB,r4)      reg(BB0, r4)     reg(BB1,r5)
reg(CC,r6)      reg(CC0, r6)     reg(CC1,r7)
reg(DD,r8)      reg(DD0, r8)     reg(DD1,r9)

/* Registers used for 2nd pass of IDCT (TMEM to PMEM) */
reg(Ap,r10)     reg(A0, r10)     reg(A1p,r11)    reg(iA1p, r18)
reg(Bp,r12)     reg(B0, r12)     reg(B1p,r13)    reg(iB1p, r16)
reg(Cp,r14)     reg(C0, r14)     reg(C1p,r15)    reg(iC1p,r16)
reg(Dp,r16)     reg(D0, r16)     reg(D1p,r17)
reg(AAp,r10)    reg(AA0p, r10)   reg(AA1p,r11)
reg(BBp,r8)     reg(BB0p, r8)    reg(BB1p,r9)
reg(C Cp,r16)   reg(CC0p, r16)   reg(CC1p,r17)
reg(DDp,r14)    reg(DD0p, r14)   reg(DD1p,r15)

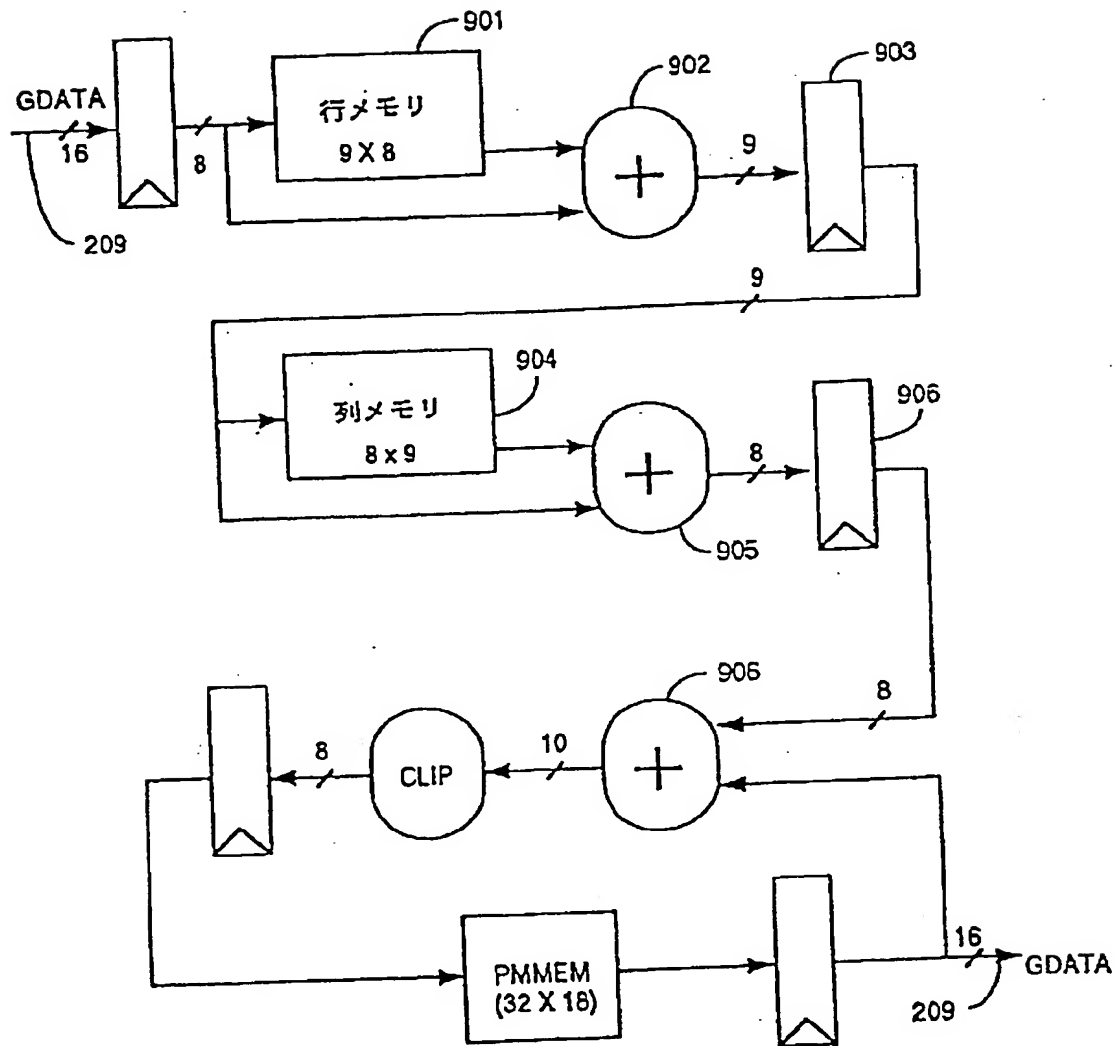
/* DEQUANTIZATION AND IDCT OF 8 x 8 BLOCK $ means operand bypasses register file */
/* % means operand is shared between 2 ops */
ori ( loopcnt, r0, 25 ) /* BUTTERFLIES MULTIPLIES */
ldmac (rT, Bp, dx0, r0, r0) /*1 Bp =b(X3p, X5p) dx0 =dmac (X0) */
dmac (nT, AA, dx3, r0, r0) /*2 AA =b(A0, B0) dx3 =dmac (X3) */
dmac (rT, Ap, dx5, r0, r0) /*3 Ap =b(X1p, X7p) dx5 =dmac (X5) */
cmac (nT, C, r0, cx2, cx6, dx4) /*4 C =b(cx2, cx6) $cx4 =cmac (dx4) */
dmac (rT, Cp, dx1, r0, r0) /*5 Cp =b(X2p, X6p) dx1 =dmac (X1) */
imac (nT, BB, r0, iA1, iB1, C1) /*6 BB =b(iA1, iB1) $iC1 =imac (C1, C0) */
dmac (BnT, D, dx7, cx0, C0) /*7 D =b(cx0, $cx4) dx7 =dmac (X7) */
imac (rT, Dp, r0, r0, r0, AA1) /*8 Dp =b(X0p, X4p) $iAA1 =imac (AA1, BB0) */
cmac (BnT, CC, cx00, D1, BB0, dx0) /*9 CC =b(D1, $iC1) cx00 =cmac (dx0) */
imac (nT, DD, r0, D0, C0, BB1) /*10 DD =b(D0, C0) $iBB1 =imac (BB1, AA0) */
imac (BwT, r0, iB1p, CC1, AA0, B1p) /*11 AAA =b(CC1, $iAA1) iB1p =imac (B1p, %BB0p) */
imac (nT, AAp, iA1p, A0p, B0p, A1p) /*12 AAp =b(A0p, %BB0p) iA1p =imac (A1p, A0p) */
imac (BwT, r0, iC1p, DD1, A0p, C1p) /*13 BBB =b(DD1, $iBB1) iC1p =imac (C1p, %C0p) */
dmac (nT, DDp, dx2, D0p, C0p) /*14 DDp =b(D0p, %C0p) dx2 =dmac (X2) */
dmac (wT, r0, dx6, DD0, AA0) /*15 DDD =b(DD0, AA0) dx6 =dmac (X6) */
cmac (nT, BBp, cx3, iA1p, iB1p, dx3) /*16 BBp =b(iA1p, iB1p) cx3 =cmac (dx3) */
cmac (wT, r0, r0, CC0, BB0, dx5) /*17 CCC =b(CC0, BB0) $cx5 =cmac (dx5) */
cmac (nT, CCp, cx1, D1p, iC1p, dx1) /*18 CCp =b(D1p, iC1p) cx1 =cmac (dx1) */
cmac (wP, r0, r0, DD0p, AA0p, dx7) /*19 DDDp =b(DD0p, AA0p) $cx7 =cmac (dx7) */
imac (BnT, B, r0, cx3, r0, AA1p) /*20 B =b(cx3, $cx5) $iAA1p =imac (AA1p, %BB0p) */
cmac (wP, r0, cx2, CC0p, BB0p, dx2) /*21 CCCp =b(CC0p, %BB0p) cx2 =cmac (dx2) */
imac (BnT, A, r0, cx1, r0, BB1p) /*22 A =b(cx1, $cx7) $iBB1p =imac (BB1p, AA0p) */
cmac (BwP, r0, cx6, CC1p, AA0p, dx6) /*23 AAAp =b(CC1p, $iAA1p) cx6 =cmac (dx6) */
imac (BnT, cx0, iB1, cx00, r0, B1) /*24 cx0 =b(cx00, 0) iB1 =imac (B1, B0) */
imac (BwP, r0, iA1, DD1p, B0, A1) /*25 BBBp =b(DD1p, $iBB1p) iA1 =imac (A1, A0) */
dmac (nT, r10, dx4, r0, A0) /*26 dx4 =dmac (X4) */
subl (loopcnt, loopcnt, 1) ) bnz (l)

```

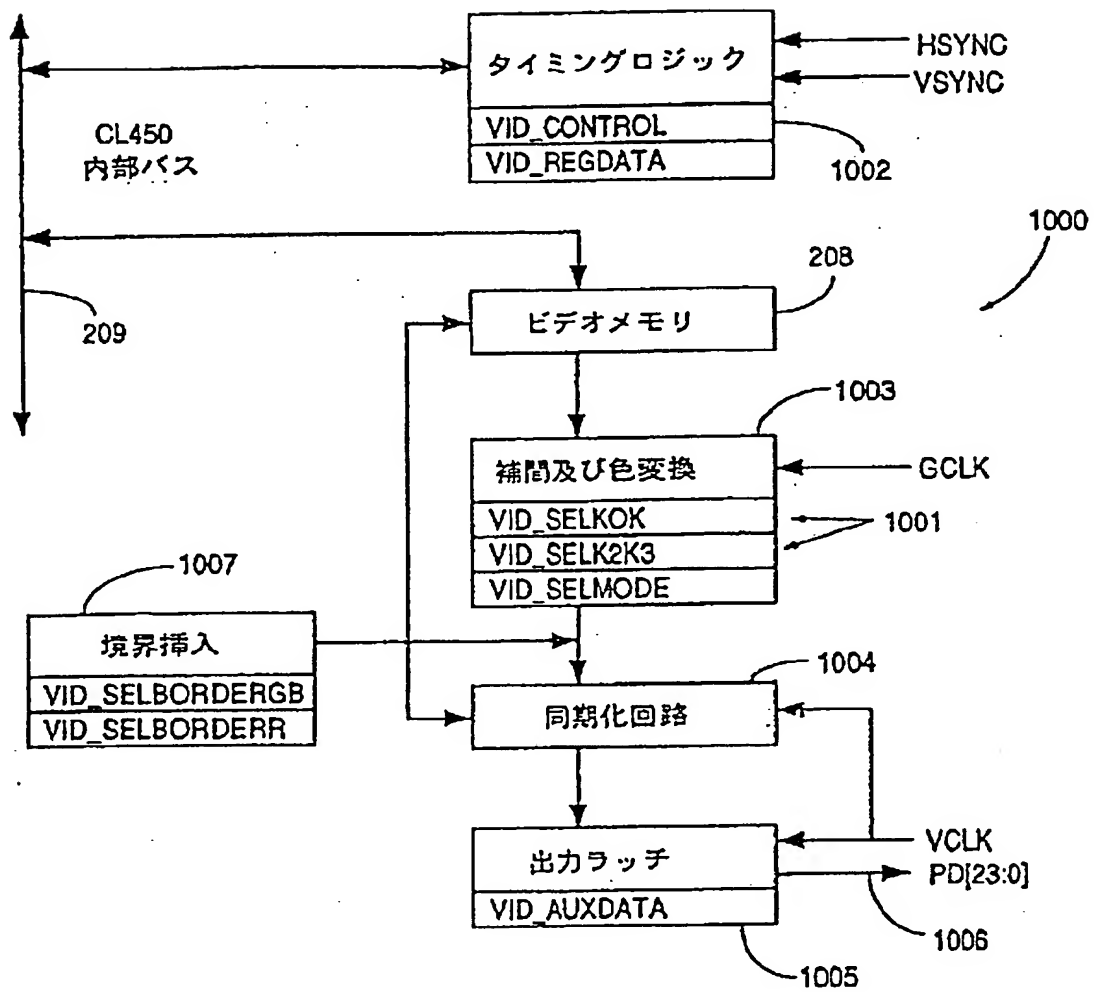
【圖 10】



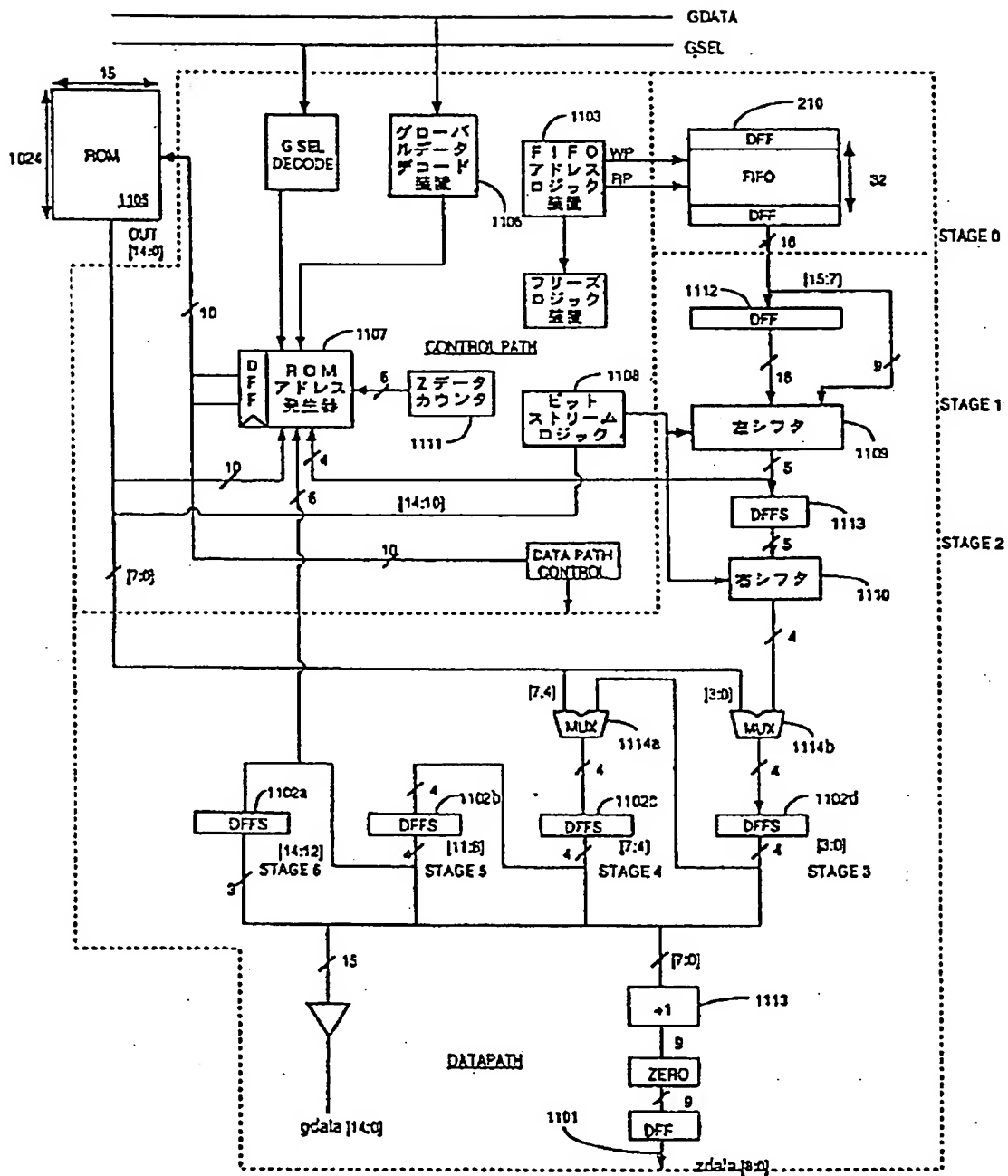
【図11】



【図 12】



【図 13】



フロントページの続き

(51) Int. Cl. 5

H 0 4 N 5/92

識別記号

庁内整理番号

D 4227-5C

F I

技術表示箇所